



CHALMERS
UNIVERSITY OF TECHNOLOGY



Schack med automatiserad pjäsförflyttning

Projektgrupp: SSYX02-17-18 Självspelande Schack

Hannes Bolmstedt – 940605

Fredric Furborg – 940413

Oskar Hulthén – 950801

Örjan Landgren – 940807

Helena Pettersson – 920224

Herman Quint - 950214

Sammanfattning

Detta projekt är ett kandidatarbete vid institutionen Elektroteknik på Chalmers Tekniska Högskola vars mål är att konstruera och bygga ett fysiskt schackspel där pjäserna kan röra sig automatiskt efter användarens eller datorns kommando. Ett schack där pjäserna kan röra sig automatiskt skulle möjliggöra att de fördelar som finns med ett digitalt schack, exempelvis möjligheten att möta en dator eller en motståndare via nätet, på ett fysiskt schackbräde. Projektet görs i utbildningssyfte och målet är inte att skapa en kommersiell produkt.

Resultatet av projektet är en prototyp vars förflyttningsmekanism av pjäser är baserad på ett stort antal elektromagneter placerade i ett rutnät under brädet. Dessa elektromagneter kan aktiveras i sekvens för att flytta de magnetiska pjäserna. Aktiveringen av elektromagneter styrs med hjälp av mikrokontrollerkortet Arduino UNO och varje elektromagnet kan aktiveras individuellt. Användarens inmatning för varje förflyttning görs i ett grafiskt schackgränssnitt.

Med prototypen går det att spela schack människa mot människa, människa mot dator eller dator mot dator. All funktionalitet har testats och även de svåraste förflyttningarna fungerar. Tyvärr är dock inte rörelserna fullt pålitliga då det ibland uppstår fel så att man behöver hjälpa till med händerna. Dessa fel kan uppstå av både mjuk- eller hårdvarubuggar, där ett exempel på det sistnämnda är att två pjäser dras ihop.

Förutom att göra pjäserörelserna pålitligare finns det många vidareutvecklingsmöjligheter för projektet såsom att implementera röststyrning, kunna flytta flera pjäser simultant eller möjlighet att använda prototypens funktionalitet till något annat än schack, förslagsvis andra brädspel.

Abstract

This project is a bachelor thesis at the institution of Electrical Engineering at Chalmers University of Technology. The main goal of this project is to design and construct a physical chessboard which enables the pieces to move automatically, after input from the user or a chess algorithm. This functionality would make it possible to face an opponent via internet or a chess computer while still using a physical board. The purpose of this project is not to create a product for commercialisation but rather to provide an interesting technical challenge for the participants of the project group.

The final result is a prototype which solves the movement of chess pieces with the help of an array of electromagnets below the chessboard. These electromagnets are not moveable but can be activated in sequence to move the magnetic pieces, one step at a time. The electromagnets are activated with an Arduino and each magnet can be activated individually. The user input is done through a graphical user interface with an integrated open source chess engine.

The prototype is finished and it is possible to play chess in three different modes: person versus person, person versus computer and computer versus computer. All functionality has been tested and even the hardest moves are possible. Unfortunately, the movements are not always performed flawlessly. This is sometimes due to software bugs, and sometimes due to hardware issue such as chess pieces clashing into each other.

There are multiple possibilities for further development, most notable would be to implement voice recognition and control, increase the reliability, implement movement of more than one piece at a time or the possibility to use the product for other applications apart from chess.

Förord

Rapporten är skriven som ett kandidatarbete vid institutionen Elektroteknik på Chalmers Tekniska Högskola. Kandidatarbetet hade en omfattning på 15 högskolepoäng och pågått under en termin. De som har gjort projektet är en blandad grupp civilingenjörstudenter från utbildningarna: Automation och Mekatronik, Datorteknik och Maskinteknik. Sammanhållning och samarbetet i gruppen har varit mycket bra, för detta är vi mycket tacksamma!

Stort tack till vår handledare Martin Fabian, som varit stöttande genom projektet. Även ett stort tack till Thomas Hammarström, som med sin tekniska kompetens varit ett bra bollplank. Ett särskilt tack till ETA - Elektroteknologsektionens Teletekniska Avdelning, där mycket av arbetet har utförts och många medlemmar bidragit med smarta idéer, komponentförslag och erfarenhet. Vi vill också rikta ett tack till Göteborgs schackcentrum som gett oss "en schackspelares perspektiv", varit positiva och uppmuntrande samt bidragit med två uppsättningar schackspel. Vidare vill vi också tacka Eric Gustafsson i XP-styret som har hjälpt oss gravera spelplanen av plexiglas till schackspelet.

Innehåll

Figurer	v
1 Inledning	1
1.1 Syfte	1
1.2 Problemformulering	1
1.3 Avgränsningar och förutsättningar	3
2 Val av förflyttningsmetod	5
2.1 Fri brainstorming	5
2.2 Undersökning av befintliga lösningar och patent	5
2.3 Genererade idéer	7
2.4 Kravspecifikation och eliminering	11
2.5 Pughmatris	13
3 Vidareutveckling av valt koncept	17
3.1 Tänkbara problem	17
3.2 Teori magnetism	18
3.2.1 Elektromagnetism	18
3.2.2 Magnetism i pjäserna	19
3.3 Magnetkonfiguration	19
3.4 Simulering av elektromagneter och pjäsers rörelse	23
3.5 Styrka på magneter och storlek på brädet	28
3.5.1 Testrigg	28
3.5.2 Test av drosslar	30
3.6 Ledmatris	31
4 Konstruktion av prototyp	33
4.1 Elektrisk koppling	33
4.1.1 Komponenter	35
4.2 Mjukvara	38
4.2.1 Arduino firmware	39
4.2.2 Java software	43
4.3 Schackbrädets konstruktion	45
4.4 Konstruktion av pjäser	47
4.5 Montering av elektronik	49
4.6 Verifiering och felsökning av hårdvaran	50
5 Resultat	53
5.1 Måluppfyllnad	54
5.2 Slutsats	56

6	Diskussion	57
6.1	Vad hade kunnat göras annorlunda	57
6.2	Förslag för fortsatt arbete	58
	Referenser	60
A	Appendix	I
A.1	Dellösningar	I
A.1.1	Igenkänning av pjäser och dess position	I
A.1.2	Inmatning för förflyttning	II
A.1.3	Konstruktion av fysiska brädet	II
A.1.4	Konstruktion av fysiska pjäserna	III
A.1.5	Extrafunktioner	III
A.2	Undersökning av andra projekt	V
A.3	Pughmatriser	VII

Figurer

1.1	Funktionsträd som visar uppdelningen av delproblem.	2
2.1	Befintlig lösning med förflyttning av en elektromagnet i två led.	6
2.2	Två exempel på befintliga lösningar.	6
2.3	Skiss på befintligt patent med robotarm som flyttar schackpjäserna.	6
2.4	Befintlig lösning med förflyttning i två led med en elektromagnet.	7
2.5	Skiss på en lösning med två centrerade armar som kan rotera och har utskjutbara elektromagneter monterade.	8
2.6	Skiss på schackbräde med klo som plockar upp och förflyttar pjäser.	8
2.7	Skiss på schackbräde med uppskjutande rutor och dolda armar som förflyttar pjäser.	9
2.8	Exempelbild på hur en liten robot som förflyttar pjäser skulle kunna se ut.	9
2.9	Liten robot som förflyttar sig med hjälp av vibrationer.	9
2.10	Små drönare i formation.	10
2.11	Rutnät av elektromagneter.	10
2.12	Whack-a-mole-spel.	11
2.13	Idéexempel, en rörlig labyrinth.	11
2.14	Kravspekifikation för slutprodukten.	12
2.15	Pughmatris med X-Y-elektromagnet (figur 2.4) som referens.	14
2.16	Slutgiltigt resultat från utvärdering av koncepten med Pughmatriser.	14
3.1	Tvärsnitt av konstruktionen som visar hur en pjäs rör sig i ett magnetfält från en avstängd elektromagnet till en aktiverad.	17
3.2	Magnetfält runt en strömförande ledare.	19
3.3	Likbenta trianglar illustrerade i koppling.	20
3.4	Lösning med 1 magnet per ruta (9 rutor).	20
3.5	Lösning med 5 magneter per ruta där rutorna delar magnet i hörnen (4 rutor).	21
3.6	Lösning med 5 magneter per ruta där de delar magnet på sidorna (4 rutor).	21
3.7	Lösning med 9 magneter per ruta (4 rutor).	22
3.8	Lösning med 25 magneter per ruta (En ruta).	22
3.9	Koncept med 9 magneter per ruta i MATLAB-simulering.	24
3.10	MATLAB-simulering steg ett.	25
3.11	MATLAB-simulering steg två.	26
3.12	MATLAB-simulering steg tre.	26
3.13	MATLAB-simulering steg sju.	27
3.14	Bilder på testriggen.	29
3.15	I bilden syns testuppsättningen med 4 drosslar av den typ som kommer att användas för schacket.	31
3.16	I tabellen presenteras de drosslar som testades, den med gul bakgrund är den drossel som användes för den slutgiltiga konstruktionen.	31

3.17	Ledmatrisen i två olika lägen.	32
4.1	Kopplingschema för en elektromagnet.	33
4.2	Kopplingschema för ihopkopplade shiftregister.	34
4.3	Mosfet med de tre benen Gate, Drain och Source.	36
4.4	MOSFET-transistor med resistor mellan jord och gaten, vilket alltså inte används.	36
4.5	Märkning och beskrivning av in-och utgångar till shiftregister 74HC595.	37
4.6	Förenklat flödesschema över Arduino-programmets kod.	40
4.7	Förenklat flödesschema över hur dödhanteringen fungerar.	41
4.8	Bilden visar hur pjäserna placeras i dödhanteringen.	41
4.9	Flödesschema över den raka rörelsen.	42
4.10	Skärmdump från det modifierade användargränssnittet för schackmotorn <i>Carballo</i>	43
4.11	Simplifierat flödesschema över Carballo med utökning som kommunicerar med Arduino-kortet.	44
4.12	Schackbrädet modellerat med träsidor.	45
4.13	CAD-modell av de tre lagren utan sammanhållande skruvar.	45
4.14	CAD-modell av de fem experimentkorten med magneter, shiftregister och Arduino UNO monterat.	46
4.15	Bild på hur graveringen ser ut, det som är svart i bilden är frostat på det färdiga brädet.	47
4.16	Bild på ett magnetiskt schackbräde och små pjäser med magneter i sig.	48
4.17	CAD-modellerade bilder på topp-delen och basen till de slutgiltiga pjäserna samt en verklig bild på en ihopsatt pjäs.	49
4.18	Montering av komponenter.	50
5.1	Tvärsnitt av konstruktionen som visar hur en pjäs rör sig i ett magnetfält från en avstängd elektromagnet till en aktiverad.	53
5.2	Två bilder som visar rutnätet av elektromagneter.	54
5.3	Bild på hela schackbrädet.	54
5.4	Kravspekifikation med resultat för protoypen.	55
A.1	Sammanställning av andra projekt och patent	VI
A.2	Pughmatris med X-Y-elektromagnet (figur 2.4) som referens	VII
A.3	Pughmatris med r- θ -elektromagnet (figur 2.5) som referens	VII
A.4	Pughmatris med klomaskin (figur 2.6) som referens	VIII
A.5	Pughmatris med robotarm (figur 2.3) som referens	VIII
A.6	Pughmatris med små radistyrda pjäser (figur 2.9) som referens	IX
A.7	Pughmatris med drönare (figur 2.10) som referens	IX
A.8	Pughmatris med elektromagneter i rutnät (figur 2.11) som referens	X
A.9	Pughmatris med vipp-labyrint (figur 2.13) som referens	X

1. Inledning

Redan under femtiotalet började datorer programmeras för att kunna spela schack [1]. Dessa datorstyrda schackmotståndare är användbara dels när man inte kan hitta en mänsklig motståndare, men har också fördelen att det går att spela med olika svårighetsgrader och att alla, nybörjare som proffs, kan utvecklas i sitt schackspel. Exempelvis var 1994 första gången en dator slog en mänsklig världsmästare då schackdatorn *Chess-Genius* slog Gary Kasparov under Intel Grand Prix i London [2].

Ett fysiskt schack är behagligare att titta på än en skärm och går att enkelt observera ur flera vinklar. Ett fysiskt schack kopplat till en dator skulle möjliggöra att de fördelar som finns med ett digitalt schack, exempelvis som möjligheten att möta en dator eller en motståndare via nätet, kan appliceras till ett fysiskt schackbräde. Ett exempel på en existerande lösning till detta problem är att användaren själv flyttar alla pjäser men att en dator talar om hur pjäserna ska placeras. Detta kan ske genom att en lampa i varje ruta indikerar vilken förflyttning som ska utföras [3]. Att användaren själv flyttar båda parter pjäser kan vara förvirrande och skapa en känsla av att användaren spelar mot sig själv.

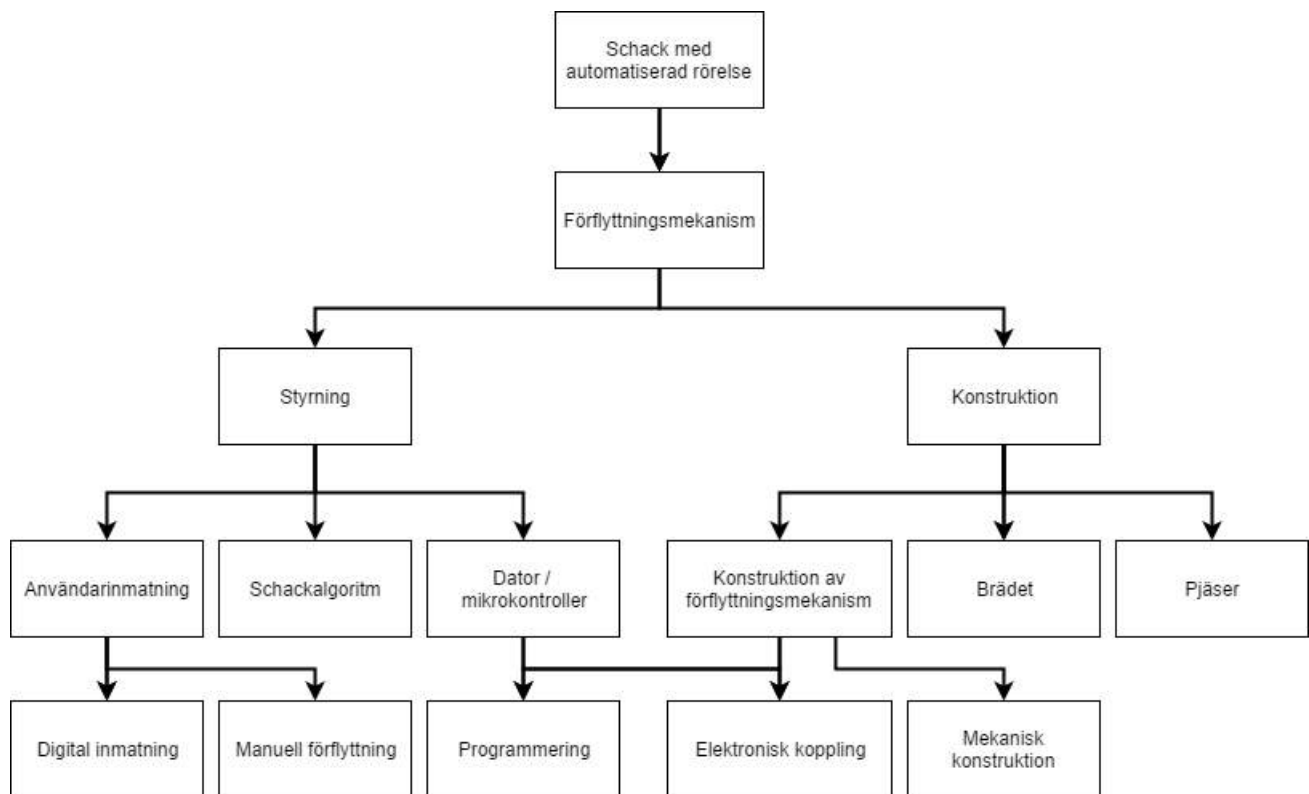
Bättre hade varit om pjäserna själva kunde flytta på sig och användaren endast behöver fokusera på sina egna drag. Möjligheten att spela på ett fysiskt bräde med automatisk pjäsförflyttning skulle dessutom kunna förbättra spelupplevelsen för personer med funktionsnedsättningar som förhindrar dem att flytta sina pjäser, då användarinamattning för drag exempelvis kan ske genom röststyrning. Tidigare lösningar på schack där pjäserna rör sig automatiskt existerar i dagsläget. Det mest kända exemplet är det fiktiva trollkarlsschacket från Harry Potter [4] vars funktionalitet detta projekt vill uppnå med teknik istället för magi. Ett exempel på en verklig befintlig produkt med denna funktionalitet är schackbrädet *Square Off* [8].

1.1 Syfte

Projektets mål är att konstruera och bygga ett fysiskt schackspel där pjäserna kan röra sig automatiskt efter användarens eller datorns kommando. Hela projektet görs i utbildningssyfte, där en utmanande teknisk lösning till ett intressant problem ska utvecklas, och inte för att skapa en kommersiell produkt.

1.2 Problemformulering

Eftersom projektet görs i utbildningssyfte och huvudproblemet med projektet redan från start är att skapa ett schack där pjäserna kan röra sig automatiskt kommer ingen djupare analys av bakomliggande problem göras. För att identifiera delproblem gjordes ett funktionsträd där utgångspunkten är huvudproblemet som sedan delas upp i mindre problem som i sin tur kan delas ytterligare. En uppdelning av detta projekt ses i figur 1.1.



Figur 1.1: Funktionsträd som visar uppdelningen av delproblem.

Ur huvudproblemet härleds förflyttningsmekanismen som det viktigaste delproblemet då resterande delproblem till stor del beror på hur förflyttningen av pjäser ser ut. Förflyttningen är sedan uppdelad i två stora delar, styrning och konstruktion.

Styrningen behöver ske med hjälp av någon form av dator och/eller mikrokontroller som behöver kunna hantera olika sorters användarinmatning. Det vill säga både digital inmatning, att användaren skriver direkt in i en dator, exempelvis på ett tangentbord, samt manuell förflyttning, att användaren flyttar sin egen pjäs. En schackalgoritm behöver också implementeras för att ge användaren möjlighet att spela mot en datormotståndare. Datorn och/eller mikrokontrollern behöver sedan programmeras och kopplas ihop med den elektroniska krets som sköter rörelsen av pjäserna.

Konstruktionen är uppdelad i tre delar. En del är konstruktionen av den mekanism som förflyttar pjäserna, en är konstruktionen och designen av själv brädet och den sista delen är konstruktionen och designen av pjäserna. I några av de initiala koncept som presenteras nedan kommer konstruktion av förflyttningsmekanism vara det samma som konstruktion av pjäser då det är pjäserna själva som står för förflyttningen. Förflyttningsmekanismen kommer troligen bestå av ett flertal elektriska och/eller mekaniska komponenter som sedan skall kopplas ihop med den dator och/eller mikrokontroller som sköter styrningen.

Nedan presenteras ett antal extrafunktioner som är önskvärda att få med i projektet i mån av tid. De är inte nödvändiga för projektet som helhet, men skulle förbättra användarupplevelsen.

- **Röststyrning**
Att spela genom att styra pjäserna med röstkommandon är ett sätt att underlätta styrningen för spelare som har svårt att ge andra kommandon eller att flytta pjäser själva.
- **Fuskindikator**
Ett tillägg som uppmärksammar spelarna om ogiltiga drag bidrar till att spelet spelas korrekt och dessutom förenklar det inläringen för nybörjare, då de får direkt feedback utifall deras drag är giltigt eller ej.
- **Schackmatt-indikator**
Syftet med en schackmatt-indikator är att uppmärksamma när spelet är klart. Genom att indikera vinsten, kan vinstkänslan förstärkas och därigenom kan även användarupplevelsen förbättras.
- **Återställning av spel**
Ofta vill spelare spela flera spelomgångar i rad. När ett spel avslutas skulle då en återställningsfunktion underlätta övergången till en ny spelomgång.
- **Effekt när pjäser dör**
Genom att använda någon form av effekt då pjäser tas ur spel skulle spelet kännas mer levande vilket bidrar till att förbättra användarupplevelsen.
- **Möjlighet att spela två spelare**
Möjligheten att kunna spela två personer mot varandra istället för att endast kunna spela mot en dator skulle förbättra användarmöjligheterna för brädet.
- **Möjlighet att spela dator mot dator**
Möjligheten att kunna observera spel och lära sig av det, eller kunna spela upp kända matcher för underhållning.

1.3 Avgränsningar och förutsättningar

För att projektet inte ska bli för stort och svårt att lösa behöver det avgränsas. Efter att undersökningar av andra befintliga lösningar genomförts, fördes diskussioner mellan gruppmedlemmar och handledare för att slutligen komma fram till följande avgränsningar och förutsättningar:

- Det kommer ej att skapas en egen schackalgoritm utan en befintlig open source schack AI (artificiell intelligens) kommer att implementeras.
- Manuell förflyttning av användarens egna pjäser kommer ej att implementeras utan all förflyttning kommer skötas automatiskt.
- Extrafunktioner, listade ovan i avsnittet 1.2, är önskvärda men ej nödvändiga. Implementeras i mån av tid beroende på komplexitet.

2. Val av förflyttningsmetod

Det främsta problemet i projektet är hur pjäserna ska flyttas. Resterande delar så som programmering, konstruktion och styrning bygger på hur förflyttningen ser ut, se funktionsträdet i figur 1.1. Eftersom valet av förflyttningsmetod låg till grund för projektet lades större vikt på att göra ett bra val för förflyttningen, än för övriga delproblem. Därför hanterades valet av förflyttningsmetod ur ett produktutvecklingsperspektiv för att metodiskt utvärdera och jämföra olika lösningar. Metoder som används byggde på sådant som gruppmedlemmar har lärt sig under tidigare produktutvecklingsprojekt och dessa metoder presenteras löpande i texten där de tillämpas.

Resterande delproblem löstes löpande där den mest lovande lösningen efter diskussion och rådgivning tillämpades. Eftersom projektet inte har som mål att kommersialisera en produkt är det acceptabelt att använda en lösning som inte nödvändigtvis är den absolut bästa så länge den fungerar och uppfyller kraven.

2.1 Fri brainstorming

Första fasen av projektet var att ta fram så många möjliga lösningar till problemet som möjligt för att senare sälla bort sämre lösningar tills bara den bästa finns kvar som i och med det vidareutvecklas. Projektet började därmed med en helt öppen brainstormingkväll där gruppen genererade idéer utan några begränsningar. Detta gjordes för att inte missa några idéer som kan komma från en ganska överklig och ogenomförbar idé. Här togs det fram idéer på hur huvudproblemet skulle lösas samt på hur några av delproblem till detta skulle kunna lösas. Alla lösningar på huvudproblemet presenteras lite senare i detta kapitel, förslag på lösningar till en del av delproblemen går att finna i appendix A.1.

2.2 Undersökning av befintliga lösningar och patent

För att efter brainstormingen få en bredare lösningsbas så undersöktes en rad olika projekt som kunde hittas på nätet och i olika patentdatabaser. Främst användes sökmotorn Google med nyckelord som “Wizard chess”, “automated chess” och “arduino chess”. Den vanligaste lösningen för liknande projekt är att flytta en elektromagnet under brädet med hjälp av servomotorer. Pjäserna är magnetiska och attraheras då till den aktiva elektromagneten och kan flyttas till önskad position. Nedan visas en bild på detta koncept, se figur 2.1 [5]. Konceptet kallas senare i texten X-Y-Elektromagnet på grund utav dess förflyttning av en elektromagnet i planet i X-och Y-led.



Figur 2.1: Befintlig lösning med förflyttning av en elektromagnet i två led.

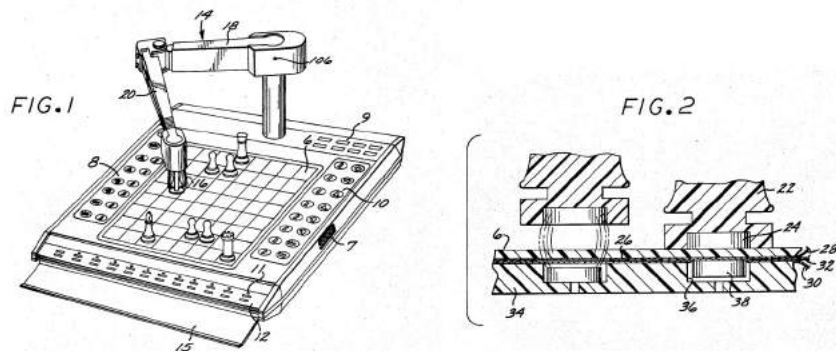
En av de tidigaste modellerna på marknaden med ett sådant förflyttningssystem var Milton Bradleys *MB Phantom* år 1982 (såld i USA under namnet *MB Grandmaster*) [6], som ses i figur 2.2a [7]. En nyare produkt med samma lösning är Kickstarter-projektet *Square Off* [8], som kommer ut på marknaden under år 2017 och kan ses i figur 2.2b [8]. Båda dessa registrerar förflyttningar genom att först trycka ner pjäsen på den ruta från vilken pjäsen ska flyttas och sedan på den ruta till vilken pjäsen ska flyttas.



(a) Det automatiska schackbrädet Milton Bradley Phantom från år 1982. (b) Det automatiska schackbrädet Square Off från år 2017.

Figur 2.2: Två exempel på befintliga lösningar.

En patenterad lösning som hittades involverar en robotarm som flyttar pjäserna. Denna arm är placerad på kanten av brädet och griper tag i en skåra i pjäserna. Varje pjäs och ruta har också varsin magnet för att pjäserna ska stå centrerat och stabilt. Denna lösning kallas senare i texten för *Robotarm* och kan ses i figur 2.3 [9].



Figur 2.3: Skiss på befintligt patent med robotarm som flyttar schackpjäserna.

Andra intressanta befintliga lösningar och patent som hanterar vissa av delproblemen finns också tillgängliga och har kort sammanfattats i en tabell som går att finna i appendix A.2. Några av dessa lösningsförslag är till exempel hur pjäserna registreras på rutorna vid en given tidpunkt, olika lösningar för schackalgoritmer, idéer på dödhantering samt en samling av schackbräden som använder sig av en schackdator.

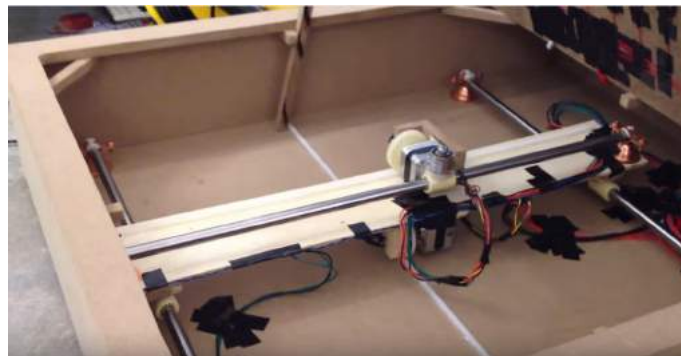
2.3 Genererade idéer

Nedan följer en sammanställning av de förflyttningskoncept som tagits fram genom idégenerering och undersökning av befintliga lösningar. De möjliga lösningarna är uppdelade i ett antal kategorier för att få en enklare översikt av alternativen, lösningar som sitter under brädet, lösningar som sitter över brädet, pjäser som rör sig individuellt, fasta lösningar utan rörliga delar och övriga lösningar som inte passar i någon av de tidigare nämnda kategorierna.

Under brädet

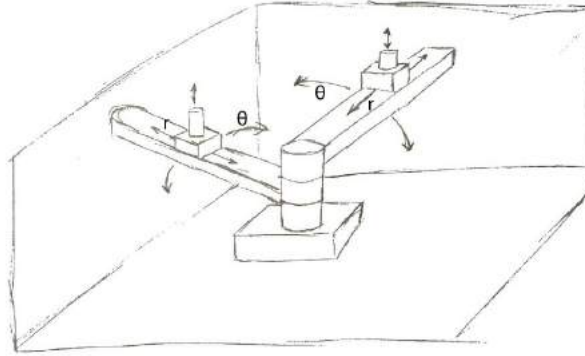
Följande lösningar som presenteras är placerade under brädet och är därmed också dolda för användaren.

- **X-Y-Elektromagnet** Under brädet kan en elektromagnet röra sig i x- och y-led med hjälp av två elektriska motorer. Pjäserna till denna lösning måste vara magnetiska och när elektromagneten aktiveras under vald pjäs attraheras pjäsen till elektromagneten, vilket gör att pjäsen dras med när den aktiva elektromagneten förflyttas. Elektromagneten samt motorerna styrs via ett mikrocontrollerkort. I figur 2.4 nedan syns en bild på detta koncept [5].



Figur 2.4: *Befintlig lösning med förflyttning i två led med en elektromagnet.*

- **r- θ -Elektromagnet** Bygger på samma princip som X-Y-lösningen ovan men använder sig av ett annat koordinatsystem. Under brädet kan en elektromagnet röra sig i r- och θ -led med hjälp av en roterande motor i mitten av brädet med utskjutande arm. Elektromagneten samt motorerna styrs via ett mikrocontrollerkort. Denna lösningen skulle kunna appliceras med flera armar samtidigt genom att sätta armarna på varandra, vilket till skillnad från X-Y-lösningen gör att flera pjäser skulle kunna flyttas samtidigt. En skiss på detta koncept ses i figur 2.5 där r- och theta-rörelsen är utritad.

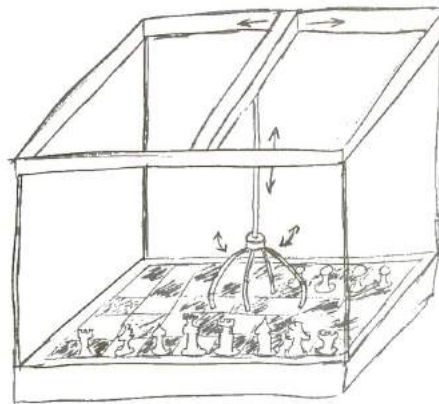


Figur 2.5: Skiss på en lösning med två centrerade armar som kan rotera och har utskjutbara elektromagneter monterade.

Ovanför brädet

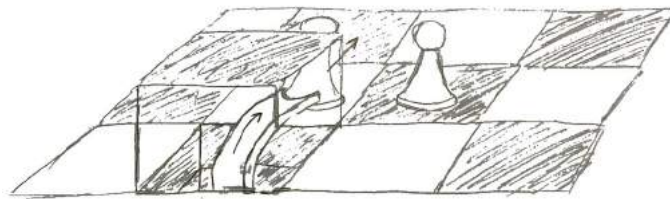
Nedan listas alla lösningar som på något sätt placeras ovanför brädet och därmed är synliga för användaren.

- **Klomaskin** Konstruktion som är inspirerad av en klomaskin med vilken gosedjur kan fångas med en klo. Konstruktionen placeras ovanför brädet och använder sig antingen av en klo som plockar upp pjäserna eller en elektromagnet som fastnar på toppen av pjäserna. En skiss på denna lösning visas i figur 2.6.



Figur 2.6: Skiss på schackbräde med klo som plockar upp och förflyttar pjäser.

- **Robotarm** En robotarm placerad i kanten av brädet som kan förflytta pjäserna. Denna lösning är redan patenterad, [9]. Flera robotarmar som är olika höga kan användas för att flytta flera pjäser samtidigt. En bild på denna lösning visas tidigare i kapitlet som figur 2.3.
- **Dolda armar** Under varje ruta finns en vridbar robotarm. När en pjäs ska flyttas öppnar sig hela tiden rutan bakom pjäsen och robotarmen under bakomliggande ruta knuffar fram pjäsen en ruta i taget. Rutorna öppnas genom att lyftas helt vertikalt och rutan ska klara att öppna sig trots att en annan pjäs står på rutan. I figur 2.7 syns en skiss på detta koncept.



Figur 2.7: Skiss på schackbräde med uppskjutande rutor och dolda armar som förflyttar pjäser.

- **Robot** En liten robot som kan åka runt på brädet och flytta pjäserna åt spelare. Roboten sköter all förflyttning själv men det behövs samtidigt något system som har kontroll på pjäsernas position som ger roboten förflyttningskommandon via trådlös kommunikation. I figur 2.8 syns ett exempel på en liten rörlig robot med en anordning som kan skjuta på pjäser [10].



Figur 2.8: Exempelbild på hur en liten robot som förflyttar pjäser skulle kunna se ut.

Pjäserna rör sig individuellt

De lösningar som följer innehåller någon slags styrning och förflyttningsmekanism i varje pjäs.

- **Små radiostyrda pjäser** Varje pjäs fjärrstyrs individuellt som en väldigt liten robot. Varje pjäs kommer behöva någon form av signalanordning för att skicka och ta emot information om hur den ska röra sig. Ett exempel på drivning är att pjäsen har hjul likt en väldigt liten radiostyrd bil. Ett annat exempel är att pjäsen drivs åt olika håll med vibrationer, ett exempel på sådan drivning kan ses figur 2.9 [11].



Figur 2.9: Liten robot som förflyttar sig med hjälp av vibrationer.

- **Drönare** Varje pjäs är en liten drönare som kommunicerar med en kommunikationscentral utanför brädet som målas på marken. Synkroniserade drönare är inget nytt [12], men det finns inga projekt att finna där det använts till att spela schack. I figur 2.10 syns en bild på små synkroniserade drönare [12].



Figur 2.10: *Små drönare i formation.*

Fasta lösningar

- **Rutnät av elektromagneter** Under brädet finns ett rutnät av fastasatta elektromagneter som drar pjäsen rätt. Närliggande pjäser som ska vara stilla får hållas fast av var sin elektromagnet för att inte dras till den aktiverade elektromagnet. Det finns sedan tidigare några lösningar där flera magneter flyttas samtidigt med denna princip [13], men som inte applicerats till schack. I figur 2.11 visas en så kallad *Boogie Box* som innehåller ett rutnät med 4x5 elektromagneter [14], denna används för att få små pappersfigurer att "dansa" genom att flytta deras magnetiska fötter mellan elektromagneter i olika sekvenser.



Figur 2.11: *Rutnät av elektromagneter.*

Övriga

- **Whack-a-mole** Pjäser dyker upp och ner ur rutorna istället för att flytta sig. Alla olika pjäser behöver identifierbart utseende. Skulle t.ex kunna styras av tryckluft där en pjäs ser olika ut beroende på hur mycket den blåses upp, exempelvis höjden av en vätskepelare eller piggar i olika formationer och höjder. Denna mekanism

måste finnas i varje ruta eftersom pjäserna inte fysiskt flyttar på sig utan bara dyker upp och ner. Konceptet är baserat på spelet whack-a-mole vilket kan ses i figur 2.12 [15].



Figur 2.12: *Whack-a-mole-spel.*

- **Vipp-labyrint**

Alla pjäser utom den som ska flyttas hålls fast av exempelvis magneter eller någon mekanisk låsning medan brädet vickar och eller vibrerar fram och tillbaka runt två axlar, vilket gör att pjäsen förflyttas genom att glida åt det håll brädet lutar åt. För att pjäserna lättare ska flytta på sig kan de vara utformade som kulor av olika storlekar och färger eftersom de hålls fast då de inte ska flytta på sig. I figur 2.13 syns en vippande labyrint som rör sig på samma sätt som detta koncept är tänkt att göra [16].



Figur 2.13: *Idéexempel, en rörlig labyrint.*

2.4 Kravspecifikation och eliminering

En kravspecifikation radar upp alla krav som finns på lösningen och som måste uppfyllas. Kraven ska vara lätta att utvärdera ifall de har uppfyllts av den färdiga lösningen. Krav sattes upp för hela slutprodukten men användes i det här fallet för att utvärdera vilka förflyttninglösningar som inte uppfyllde kraven. Kravspecifikationen ses i figur 2.14 nedan.

Kravspecifikation		
Krav	Mål	Verifikationsmetod
1. Prestanda		
a. Tid för en pjäs förflyttning	<5 s	Tidtagning för det längsta draget
b. Svårighetsgrad	>1500 ELO	Kontrollera specifikationer för schackalgoritmen
c. Ljudvolym	<60 dB (Under normal samtalsvolym)	Mätning av ljudnivå.
d. Elförsörjning	Klara svenskt eluttag	Testa att koppla in det i ett eluttag
e. Placering inom rutor	Pjäsen ska vara inom ramarna för varje ruta	Kontrollera att alla pjäser står innanför rutorna under 5 schackpartier
f. Förflyttning	Pjäsförflyttning ska utföras utan att slå i andra pjäser	Kontrolleras under 5 schackpartier
g. Identifierbarhet	Alla pjäser ska vara enkla att identifiera	Fråga person utanför projektgruppen om den kan identifiera pjäserna
2. Storlek		
a. Brädets volym	<1x1x1m	Mät prototyp
b. Pjäsens volym	Alla pjäser ska få plats på brädet	Kontrollera att alla pjäser får plats vid startposition av spel
3. Vikt		
a. Vikt för hela prototypen	<20 kg	Väg prototypen
b. Vikt för en pjäs	<100 g	Väg pjäserna
4. Pris		
Totalkostnad för projektet	<10 000 kr	Beräkna kostnad av samtliga inköp
5. Övrigt		
a. Realisbar inom tidsramarna	Fungerande prototyp innan arbetet är slut.	Testa om det går att spela schack med prototypen

Figur 2.14: Kravspecifikation för slutprodukten.

De lösningar som eliminerades efter avstämning mot kravspecifikationen listas nedan med tillhörande förklaring till varför de togs bort:

- **Dolda armar** (ses i figur 2.7) Enligt kravspecifikationen får en förflyttning max ta 5 sekunder att utföra. Denna lösning kräver förflyttning i flera avancerade steg som med stor sannolikhet skulle ta lång tid vid längre drag, vilket i sin tur gör att den inte lever upp till kraven.
- **Robot** (ses i figur 2.8) Roboten får problem då pjäser är omringade. Den kommer då att behöva flytta på pjäser som är i vägen vilket gör att dragen kommer ta mer än 5 sekunder och därmed elimineras lösningen. Om brädet är större kan roboten röra sig mellan pjäser men då ökar också avståndet för förflyttningen och tiden blir återigen kritisk.
- **Whack-a-mole** (ses i figur 2.12) Pjäserna måste lätt kunna urskiljas från varandra och en spelare måste snabbt och utan problem kunna känna igen varje pjäs. Det anses svårt att uppnå här och därför elimineras denna lösning.

Kvarstående förflyttningslösningar för vidare utvärdering:

- **X-Y-Elektromagnet** (ses i figur 2.4)
- **r- Θ -Elektromagnet** (ses i figur 2.5)
- **Klomaskin** (ses i figur 2.6)
- **Robotarm** (ses i figur 2.3)
- **Små radiostyrda pjäser** (ses i figur 2.9)
- **Rutnät av elektromagneter** (ses i figur 2.11)
- **Vipp-labyrint** (ses i figur 2.13)

2.5 Pughmatris

En Pughmatris är ett verktyg för att utvärdera lösningar mot varandra och se vilken som är bäst uppfyller kraven. För det här projektet användes Pughmatriser endast för utvärdering av alla lösningar på huvudproblemet förflyttning. Pughmatrisen fungerar på sådant sätt att alla lösningar listas upp på ena axeln och på den andra axeln kriterier mot vilka lösningarna utvärderas. För varje kriterie finns en vikt som indikerar hur viktigt just det kriteriet är. Utefter detta utvärderas varje koncept mot en referenslösning, som bestäms på förhand, genom att gå igenom varje kriterie steg för steg och bestämma om nästa lösning är bättre eller sämre än referenslösningen. Är den bättre får den 1, är den sämre får den -1 och är den lika bra får den 0. Dessa värden läggs sedan ihop (nettovärde) tillsammans med varje kriteries vikt (viktat nettovärde) och tillslut fås ett resultat på hur mycket bättre eller sämre varje lösning är jämfört med referenslösningen (rangordning). Varje lösning används som referenslösning en gång för att få en rättvis bedömning av alla koncept. Därefter läggs alla viktade nettovärden ihop till en total efter vilken lösningarna rankas.

De kriterier som jämfördes i detta projekt var följande:

- **Förflyttningshastighet**
Ett drag bör inte ta för lång tid. En hög förflyttningshastighet är därför mycket viktigt. Vikt: 5.
- **Precision**
Det är ett krav att pjäser alltid placeras helt inom rutorna. Det är önskvärt men ej av stor vikt att de placeras så nära mitten som möjligt. Vikt: 2.
- **Enkelt att programmera**
Projektet ska vara så enkelt att genomföra som möjligt när det gäller mjukvara. Dock inte av högsta vikt så länge det anses genomförbart. Vikt: 3.
- **Enkel tillverkning**
Projektet ska vara så enkelt att genomföra som möjligt när det gäller hårdvara. Projektet är dock tidsbegränsat och verkstadsresurserna är begränsade. Vikt: 2.
- **Vikt**
Det färdiga schacket skall vara portabelt och gärna så lätt som möjligt. Vikt: 2.
- **Storlek**
Det färdiga schacket skall vara portabelt och kunna förvaras. Det behöver därmed vara av rimlig storlek. Vikt: 2.
- **Multi-movement**
Multi-movement innebär förmåga att kunna flytta flera pjäser samtidigt vilket kan tänkas användas då pjäser slås ut, återställs eller vid rokad. Det förbättrar också möjligheterna att det färdiga konceptet kan användas för andra applikationer än schack, exempelvis andra sällskapsspel. Vikt: 3.

- **Användarupplevelse**

Detta är svårt att utvärdera då det är helt subjektivt. Diskussion mellan gruppens medlemmar är grunden för hur koncepten jämförts med avseende på användarupplevelse. En av anledningarna till att göra ett digitalt schack fysiskt är att förbättra användarupplevelsen och det är därför viktigt att detta uppfylls. Vikt: 5.

- **Ljudnivå**

Eventuella oljud från schackspelet sänker användarupplevelsen. Lägre ljudnivå är därför att föredra. Vikt: 2.

- **Kostnad**

Så länge kostnaden ej överstiger budgeten och är motiverad är det ej av stor vikt, då projektet inte är tänkt att kommersialiseras utan mest är en teknisk utmaning. Vikt: 1.

Då arbete med Pughmatris är en stor process med mycket data presenteras endast första steget i processen figur 2.15 samt resultatet figur 2.16, resterande steg finns att finna i appendix A.3.

Chalmers		Pughmatris 1 (Relativ beslutsmatris):							
Uttfärdare:									
Fredric Furborg Herman Quint Oskar Hulthén		Skapad: 170216							
Örjan Landgren Helena Pettersson		Modifierad: 170221							
Kriterier		Alternativ							
	Vikt	XY-Magnet	r-8 -Magnet	Klomaskin	Robotarm	Små radiostyrda pjäser	Drönare	Elektromagnet er i rutnät	Vipp-labyrint
Förflyttningshastighet	5		0	-1	1	0	1	1	-1
Precision	2		0	0	0	-1	-1	-1	-1
Enkelt att programmera	3		-1	0	-1	-1	-1	-1	-1
Enkel tillverkning	2		-1	-1	-1	-1	1	-1	-1
Vikt	2		0	-1	-1	-1	1	-1	-1
Storlek	2		0	-1	-1	1	-1	0	-1
Multi-movement	3		1	0	1	1	1	1	0
Användarupplevelse	5		0	-1	-1	1	1	1	1
Ljudnivå	2		0	0	0	0	-1	1	-1
Kostnad	1		0	0	0	-1	-1	-1	-1
Nettovärde	27	0	-1	-5	-3	-2	0	-1	-7
Viktat nettovärde		0	-2	-16	-6	0	7	5	-14
Rangordning		3				3	1	2	

Figur 2.15: Pughmatris med X-Y-elektromagnet (figur 2.4) som referens.

	XY-Magnet	r-8 -Magnet	Klomaskin	Robotarm	Små radiostyrda pjäser	Drönare	Elektromagnet er i array	Vipp-labyrint
Totalt viktat nettovärde:	26	12	-45	-17	14	37	57	-84
Total rangordning:	3	5	7	6	4	2	1	8

Figur 2.16: Slutgiltigt resultat från utvärdering av koncepten med Pughmatriser.

Ur resultatet från Pughmatrisen eliminerades direkt de tre klart sämsta lösningarna: Vipp-labyrint, Klomaskin och Robotarm. Kring övriga lösningar hölls en diskussion inom gruppen i samarbete med projektets handledare. Koncepten små radiostyrda pjäser samt drönare eliminerades på grund av att dessa, vid diskussion, ansågs vara för komplexa och skulle ta alldeles för lång tid att genomföra.

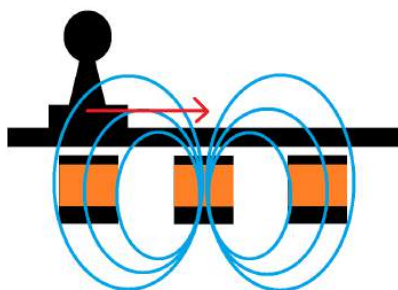
X-Y-magnetlösningen har genomförts flertalet gånger i andra projekt. I diskussion med handledaren fördes resonemanget att det redan är känt att lösningen fungerar och att det finns mer utmaning och forskningssyfte i att testa en annan ny lösning. Dessa argument i kombination med att en del av projektets syfte är att genomföra projektet i utbildnings-syfte gjorde att det kändes mer relevant ur ett ingenjörsperspektiv att hitta nya vägar och därför valdes denna lösningen bort.

De övriga två lösningarna är då r - Θ -lösningen och ett rutnät av elektromagneter. r - Θ -lösningen är väldigt lik X-Y-magnetlösning och samma resonemang kan delvis användas för att eliminera denna. Den är också klart sämre än ett rutnät av elektromagneter enligt Pughmatrisen och valdes således bort.

Den lösning som valdes för fortsatt utveckling var därför ett rutnät av elektromagneter, ett nytt koncept med nya tekniska utmaningar. Eftersom konceptet inte har använts för schack tidigare behövdes en djupare analys av funktionen göras innan produkten konstrueras. Nästa kapitel hanterar de frågeställningar som behövde besvaras för att kunna använda den valda konceptidén som ett schackbräde.

3. Vidareutveckling av valt koncept

Konceptidén som valdes går ut på att det sitter ett rutnät av elektromagneter under schackbrädet. Varje pjäs måste innehålla något magnetiskt ämne och ska på så sätt kunna flytta sig mellan elektromagneterna. För att utföra en förflyttning sätts elektromagneterna på i sekvens som gör att pjäsen rör på sig från sin startposition, en elektromagnet i taget, tills pjäsen når sin slutposition. Detta illustreras i figur 3.1, där de blå linjerna motsvarar ett magnetfält som kommer från den aktiva elektromagneten i mitten vilket gör att pjäsen som är över en intilliggande elektromagnet attraheras och förflyttas till den nya positionen (röd pil). Detta koncept bygger på det tidigare nämnda lösningsförslaget, rutnät av elektromagneter, som finns beskrivet i avsnitt 2.3.



Figur 3.1: Tvärsnitt av konstruktionen som visar hur en pjäs rör sig i ett magnetfält från en avstängd elektromagnet till en aktiverad.

3.1 Tänkbara problem

Det största problemet är att veta vad som krävs för att kunna förflytta en pjäs från en elektromagnet till en intilliggande elektromagnet. Dessutom finns det en mängd olika typer av förflyttningar inom schack som gör att det kan bli krångligt när det kommer till utformningen av rutnätet. Några av de mest komplicerade förflyttningarna är till exempel när hästen rör sig först ut i en ny omgång eller att en helt omringad pjäs slås ut (de rörelser som vanligen kräver att pjäserna måste lyftas). Som helhet innebär detta att ett tänkbart problem för förflyttningen är när pjäserna ska “hoppa över” varandra och samtidigt ha kontakt med brädet.

Det som huvudsakligen behövde undersökas var alltså svar på dessa tre frågor angående förflyttningen:

1. Hur många elektromagneter krävs?
2. Hur starka behöver elektromagneterna vara?
3. Hur nära varandra ska elektromagneterna vara?

Dessa frågor behandlas löpande i efterföljande text, och för att lägga en grund till att kunna svara på dessa frågor behövs lite bakomliggande teori om hur magnetism fungerar.

3.2 Teori magnetism

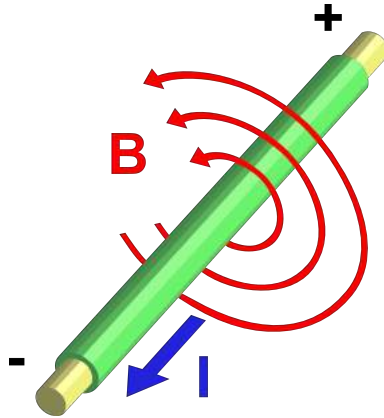
Konceptet bygger på användningen av elektromagneter och magneter samt hur dessa dimensioneras och möjligen också tillveckas, därmed har en studie kring magnetism och elektromagnetism utförts.

Magnetism orsakas av att elektriska laddningar rör på sig. I en permanentmagnet orsakas magnetfältet av elektronernas rörelse i atomerna och eftersom dessa riktas på ett sådant sätt att de samverkar bildar de en magnet [17]. En magnet har en nordpol och en sydpol, lika poler repellerar varandra medan olika poler attraheras till varandra. För att detta ska ske och alltså orsaka kraftverkan krävs det minst två magneter som interagerar.

3.2.1 Elektromagnetism

Runt strömförande ledare bildas magnetfält, på grund av elektronerna som rör sig i ledaren. Detta kan observeras genom att hålla en kompassnål (en liten permanentmagnet) bredvid en ledare som först inte leder ström och sedan koppla in ledaren så att den leder ström. När strömmen sätts på kommer kompassnålen byta position på grund av magnetfältet från ledaren. De fjärrverkande krafter som uppstår runt och mellan magneter beskrivs med magnetiska fält som kan liknas vid elektriska fält. Magnetfältet som bildas runt en ledare illustreras i figur 3.2, där de röda pilarna visar magnetfältets riktning och den blå pilen visar strömmens riktning. Magnetfältets styrka är proportionerligt mot strömmen. Hur starkt ett magnetfält är mäts i magnetisk flödestäthet, vilket betecknas som B i figur 3.2.

Om en ledare formas som en tät spole, vilket kallas för en solenoid, kommer det magnetfält som bildas runt ledaren att samverka och bilda ett större magnetfält runt hela spolen. Magnetfältet i en solenoid blir starkare för varje varv som spolen har. Skulle spolen dessutom pressas ihop på ett sådant sätt att dessa varv ligger tätt, förstärks magnetfältet ytterligare. Genom att placera en kärna med hög permeabilitet i centrum av spolen kan magnetfältet ytterligare förstärkas. Permeabilitet innebär förmågan hos ett material att uppehålla ett magnetiskt fält, typiskt material med hög permeabilitet är ferrit som är en typ av järn med en viss kristallstruktur. En skärmad elektromagnet har en järnkärna som dessutom går utanför spolen och helt innesluter den, vilket leder till ett ännu starkare magnetfält, men som är inneslutet i järnet. Eftersom elektromagneterna i projektet ska kunna attrahera pjäser krävs det att magnetfältet sprids ut och därför behövs oskärmade elektromagneter.



Figur 3.2: Magnetfält runt en strömförande ledare.

3.2.2 Magnetism i pjäserna

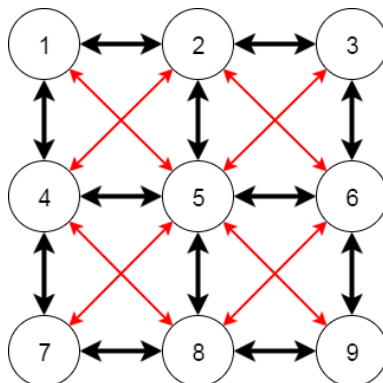
För att en schackpjäs ska reagera på magnetfältet som skapas av en aktiv elektromagnet krävs att materialet är magnetiskt. En pjäs helt gjord i plast eller trä hade därför inte fungerat. Man kan dock tänka sig att man använder sig av någon metall med hög permabilitet som till exempel ferrit. Trots att ferrit eller andra magnetiska metaller inte är magnetiska av sig själv kan de bete sig som magneter i närheten av magnetiska fält. Detta eftersom det magnetiska fältet ordnar de annars olikriktade magnetiska krafterna i järnbiten på ett sådant sätt att järnbiten blir magnetisk under tiden som den utsätts för magnetfältet.

Ett annat alternativ skulle vara att använda en permanentmagnet, som har låg permabilitet, men vars egna poler skulle bidra till en större attraktionskraft. Kortfattat bör pjäserna vara lätta och ha något magnetiskt material i botten. Fördelen med att använda en permanentmagnet är att det gör att elektromagneterna inte behöver vara lika starka för att flytta på pjäsen. Nackdelen är att även när elektromagneten är avstängd så dras pjäsen, ganska starkt till elektromagnetens kärna. För att den attraherande kraften inte ska vara allt för stark krävs det att avståndet ökas mellan brädet och elektromagneterna. Detta leder i sin tur till att elektromagneterna måste ha hög spridning på magnetfältet på ett sådant sätt att det når pjäserna och bibehåller önskad förflyttningsförmåga. I detta projekt observerades att det krävdes ett betydligt starkare magnetfält för att kunna förflytta en bit ferrit jämfört med permanentmagneter och såldes valdes det sistnämnda att användas i pjäserna.

3.3 Magnetkonfiguration

Innan det går att ta reda på hur starka elektromagneterna ska vara och hur nära de ligger varandra behöver deras placeringsmönster (konfiguration) designas och då som följd även antalet elektromagneter. När det kommer till hur många elektromagneter som krävs för att konceptet ska fungera som tänkt, togs fem tänkbara lösningar fram och dessa behandlas nedan. Tanken bakom hur det ska fungera är att varje likbent triangel som skapas mellan magneterna (bildas av en röd och två svarta pilar i figur 3.3) får betydelse för förflyttningen. Några av lösningarna är beroende av att magneterna bara drar pjäserna längs de korta sidorna av triangeln (svarta pilarna i figurerna) och att pjäserna inte påverkas längs hypotenusan i triangeln (röda pilarna i figurerna). Detta medför att elektromagneterna behöver dimensioneras precis för att det ska kunna fungera. Om längden mellan

magneterna (katet) beskrivs som a kommer hypotenusan, här betecknad som b , bli ca 41% längre. Detta visas matematiskt i ekvationen: $b^2 = a^2 + a^2 \rightarrow b = \sqrt{2} \cdot a \approx 1.41 \cdot a$. Elektromagneterna måste därför dimensioneras till att klara av att dra pjäser på avståndet a men inte avståndet $1.41a$ i de lösningsförslag där det spelar roll.



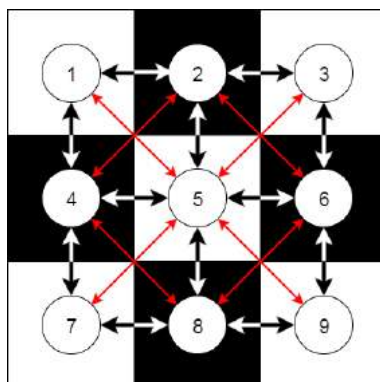
Figur 3.3: Likbenta trianglar illustrerade i koppling.

För att kunna bestämma vilken magnetkonfiguration som är bäst togs en del idéer fram genom logiska resonemang. Dessa idéer genererade i fem tänkbara lösningar som presenteras och ges en kortfattad bedömning nedan.

- **1 Elektromagnet per ruta**

Det intuitiva fallet av att ha en magnet per ruta, se figur 3.4, medför att man behöver flytta runtliggande pjäser för att kunna utföra många av dragen. Framförallt bildas en kedjereaktion då en en pjäs ska flyttas till en ruta som angränsar till en annan pjäs. Då behöver magneten under denna angränsade pjäs aktiveras, och alla magneter under pjäser som angränsar till denna och så vidare. Problem uppstår också för omringade pjäser då alla pjäser som står i vägen måste flyttas innan draget kan utföras. Detta ansågs kunna skapa förvirring i var pjäserna egentligen ska stå och komplicerar programmeringen avsevärt, vilket ledde till att lösningar med flera elektromagneter per ruta utforskades.

Att använda sig av en elektromagnet per ruta skulle resultera i att brädet, för 8x8 rutor endast skulle behöva 64 magneter totalt.

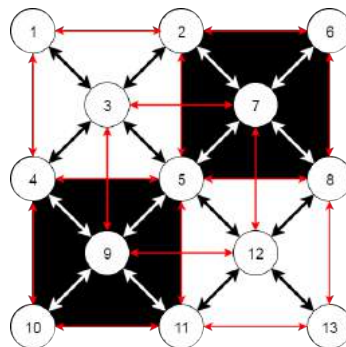


Figur 3.4: Lösning med 1 magnet per ruta (9 rutor).

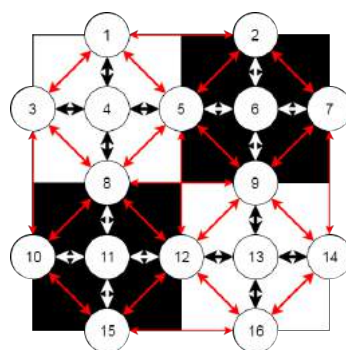
- **5 Elektromagneter per ruta**

Att lägga till magneter i varje ruta som delar hörn, se figur 3.5, eller att göra så magneterna delar sida, se figur 3.6, skulle möjliggöra bättre förflyttning än i lösningsförslaget med en magnet per ruta. Den lösning som delar hörn kan lösa den **diagonala** rörelsen utan att någon annan pjäs än den som faktiskt ska röra på sig flyttas. Samtidigt kan den lösning som delar sina sidor klara av den **raka** rörelsen utan att någon annan pjäs behöver flytta på sig. Problemet med bägge dessa lösningar är att man inte kan röra sig längs med sidorna på rutorna vilket innebär att rörelser som hästens eller utslagningens skulle kräva att många pjäser behöver flyttas undan för att ge plats åt förflyttningen. Dessutom skulle den motsatta rörelsen, exempelvis den raka rörelsen i lösningsförslaget som delar hörn både behöva gå sick-sack över brädet och flytta undan andra pjäser.

Dessa lösningsförslag innebär som nämnt att andra pjäser fortfarande behöver flyttas vid vissa rörelser. Till skillnad från det tidigare lösningsförslaget med en elektromagnet per ruta då pjäserna behövde flyttas till nästliggande rutor så kan pjäserna med dessa förslag istället temporärt flyttas till hörn/kanter inom sin egen ruta. Även om de temporärt flyttade pjäserna befinner sig inom sin ursprungliga ruta ansågs det inte optimalt att behöva flytta någon annan pjäs än de som faktiskt ska flyttas. Lösningen av att ha elektromagneterna orienterade som ett kors i rutan, se figur 3.6, skulle innebära en användning av 208 elektromagneter för brädet med 8x8 rutor. Däremot skulle lösningen med elektromagneterna orienterade som ett kryss, se figur 3.5, innebära att brädet med 8x8 rutor utgörs av endast 145 elektromagneter.



Figur 3.5: Lösning med 5 magneter per ruta där rutorna delar magnet i hörnen (4 rutor).

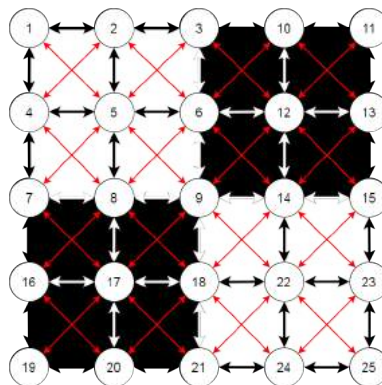


Figur 3.6: Lösning med 5 magneter per ruta där de delar magnet på sidorna (4 rutor).

- **9 Elektromagneter per ruta**

Att lägga till ytterligare elektromagneter på ett sådant sätt att varje ruta innehåller 9 magneter som delar kanter med sina intilliggande rutor, se figur 3.7, skulle möjliggöra rörelse emellan redan upptagna rutor. Denna rörelse skulle innebära att ingen annan pjäs behöver flyttas undan vid en given pjäs rörelse, förutsatt att elektromagneterna är såpass precist dimensionerade att pjäserna ej kan röra sig över det avstånd som är illustrerat med röda pilar i figur 3.7.

Detta lösningsförslag skulle behöva totalt 289 elektromagneter. Ytterkanterna kan tas bort då pjäserna inte längre behöver flytta sig till sina yttre positioner för att göra plats åt andra pjäser som förflyttas, vilket resulterar i totalt 225 elektromagneter.

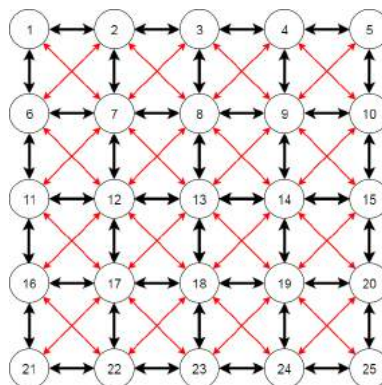


Figur 3.7: *Lösning med 9 magneter per ruta (4 rutor).*

- **25 Elektromagneter per ruta**

För att lösa problemet med att ha en jämn rörelse för både diagonal och rak förflyttning skulle antalet elektromagneter behöva skalas upp till hela 25 stycken. Detta skulle medföra att brädet snarare skulle dimensioneras till att dra för både de svarta och de röda pilarna i figur 3.8, då det alltid kommer finnas minst en elektromagnet emellan vid varje rörelse oavsett riktning. Att hypotenusan av varje triangel drar skulle medföra att den diagonala rörelsen över brädet kan bli jämn. Negativt är att denna lösning kräver en väldigt stor mängd elektromagneter samt att rutorna skulle behöva vara relativt stora då 5x5 elektromagneter ska få plats.

Den mängd elektromagneter som skulle krävas för denna lösning för 8x8 brädet är 1089st, varav de två yttre raderna kan tas bort, vilket resulterar i totalt 845 elektromagneter.



Figur 3.8: *Lösning med 25 magneter per ruta (En ruta).*

- **Val av konfiguration** När de fem tänkbara lösningarna ställs emot varandra är lösningarna med 9 respektive 25 elektromagneter mest lovande. Lösningen med att ha 25 magneter förkastas dock på grund utav att antalet magneter för att jämna ut den diagonala rörelsen “kostar” cirka 620 elektromagneter, vilket hade gjort schackbrädet både betydligt större och dyrare. Därför vidareutvecklades lösningsförslaget med att ha 9 magneter per ruta och detta främst eftersom att antalet magneter är rimligt med näst intill full funktionalitet.

En sak att notera är att det maximala antalet aktiva elektromagneter vid en specifik rörelse begränsas av strömkällan som driver elektromagneterna. För en rörelse i lösningen med 9 magneter är det maximala antalet elektromagneter som är på samtidigt 3 stycken (då en pjäs rör sig emellan två upptagna rutor), samtidigt i lösningen med 25 magneter är det maximala antalet endast 1 aktiv elektromagnet per rörelse. Att minimera antalet aktiva elektromagneter för varje rörelse skulle möjliggöra att fler rörelser kan tillåtas ske samtidigt.

Förutom de 15x15 elektromagneter som krävs för att realisera lösningen med 9 magneter per ruta, läggs 2 kolumner till per sida för att hantera utslagna pjäser. Endast 15 utslagna pjäser per sida krävs då kungen aldrig slås ut i schack. Därmed blir det 19 kolumner och 15 rader med elektromagneter, det vill säga totalt $15 \cdot 19 = 285$. Hur dödhanteringen fungerar går att läsa mer om i avsnitt 4.2.1, angående Arduino firmware.

3.4 Simulering av elektromagneter och pjäsers rörelse

För att testa logiken bakom rörelsen för olika magnetkonfigurationer har en modell tagits fram i MATLAB. Kraften genererad av elektromagneterna i modellen bygger på Maxwells dragkraftsformel [18] vilken visas nedan:

$$F = \frac{(NI)^2 \mu_0 A}{2L_{core}^2} \quad [N]$$

$N = 500$, antal varv i spolen

$I = 1$, strömmen i ampere [A]

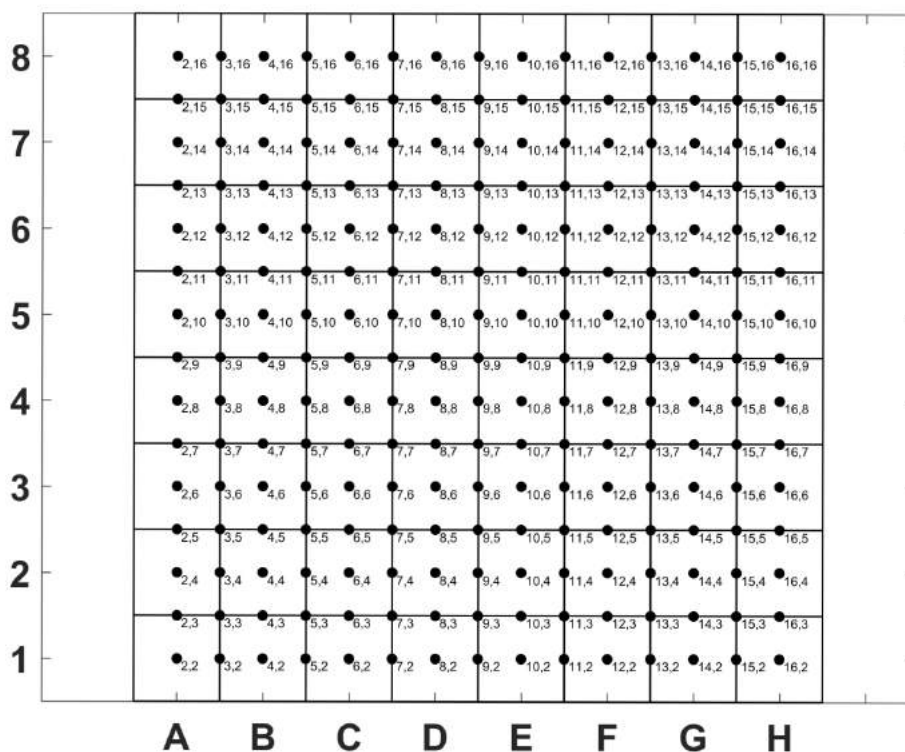
$A = 0.08^2$, tvärsnittsarean hos kärnan [m^2]

$L_{gap} =$, avståndet i meter, mellan magnetspolen och pjäsen ('luftgapet')

$\mu_0 = 4\pi \cdot 10^{-7}$, permeabiliteten hos rymden (luft)

Detta är en dålig approximation då den kräver att kärnan är en nästan komplett loop där luftgapet är litet i förhållande till magnetfältets tvärsnitt. Simuleringen används dock främst till att testa logiken och ligga till grund för programmeringen av mikrokontrollerkortet, absoluta värden är därför inte nödvändigt. För logikens skull krävs endast att magnetfältet är rotationssymmetriskt kring z-axeln samt att magnetfältets styrka avtar kvadratisk med avståndet.

I följande exempel är magneterna placerade enligt konceptet med 9 magneter per ruta, se figur 3.9. De svarta punkterna motsvarar elektromagneter där avståndet mellan magneterna är 3 centimeter, de små siffrorna i figuren är för att indexera magneterna.

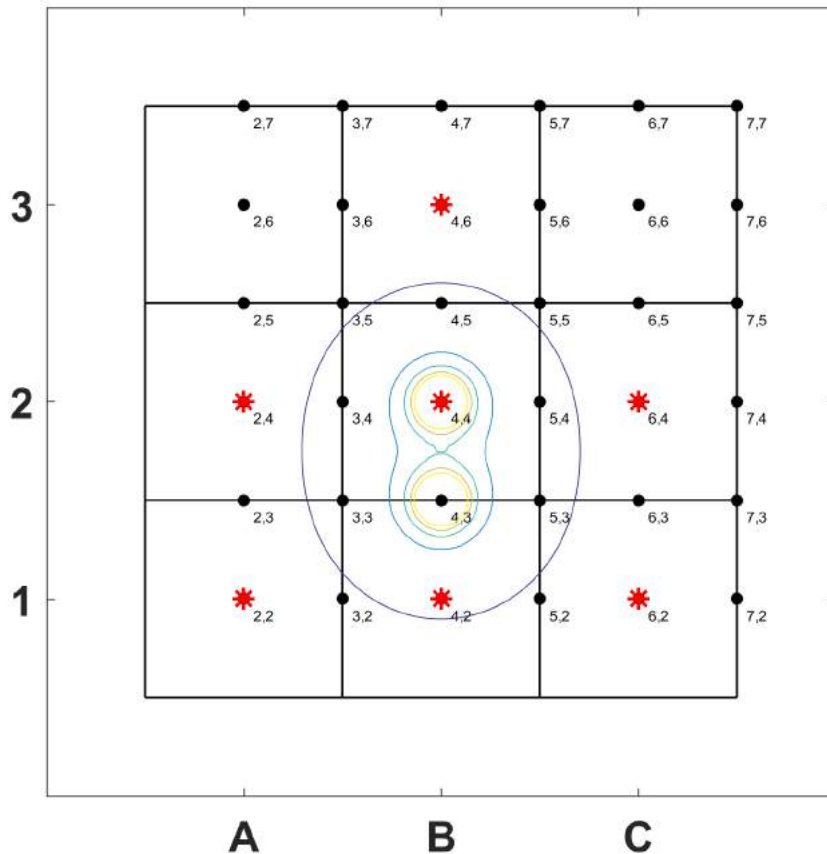


Figur 3.9: *Koncept med 9 magneter per ruta i MATLAB-simulering.*

För beskriva hur simuleringen är uppbyggd och visa att konceptet med 9 magneter per ruta fungerar kommer ett exempel visas. Börja med att studera startpositionen i figur 3.10. De röda stjärnorna motsvarar pjäser. Konturlinjerna motsvarar kraftfält från aktiva elektromagneter, där den yttre linjen motsvarar den kritiska kraften för att en pjäs ska flytta på sig. Denna kraft härstammar från friktion samt att magnetiska pjäser attraheras till kärnan av elektromagneterna och har i simuleringen uppskattats till 1 Newton.

I detta exemplet skall en häst flyttas från B1 till C3 vilket innebär att den är helt instängd och måste ta sig mellan och förbi flera andra pjäser. Om denna förflyttning är möjlig tyder det på att konceptet med 9 rutor per ruta också är det. Givetvis testades också flera andra rörelser i simuleringen vilka inte gås igenom här.

Först beräknas den optimala vägen för förflyttningen med hjälp av algoritmen A* vilken också behöver ta hänsyn till andra pjäser och gå runt dessa. A* är en välkänd algoritm för att lösa 'kortaste vägen'-problem. För detta projekt användes ASTARPATH från MATLAB file exchange, gjord och upplagd av Einar Ueland [19], vilken har modifierats för att endast tillåta horisontell och vertikal förflyttning. Den väg som algoritmen hittade går att se i figur 3.13 där den röda linjen motsvarar den väg som pjäsen flyttats.

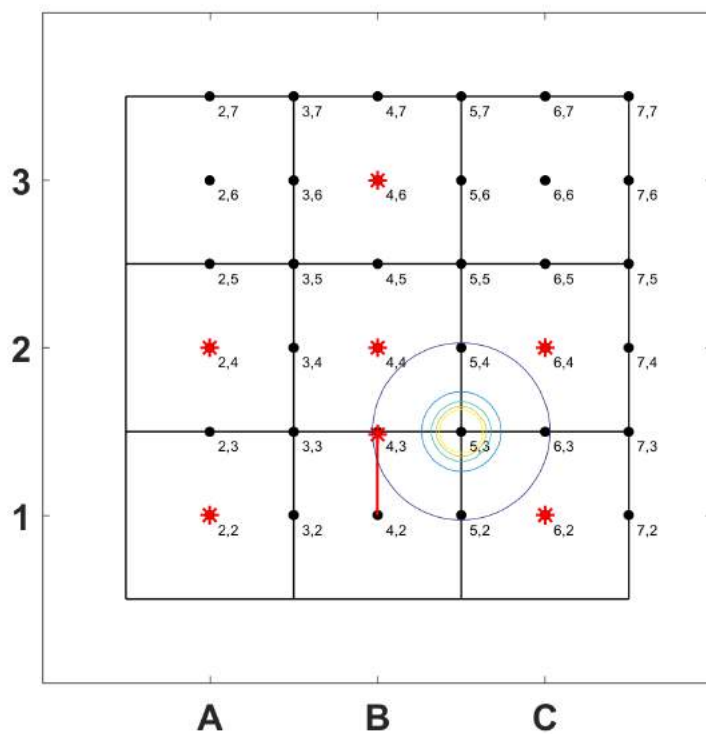


Figur 3.10: *MATLAB-simulering steg ett.*

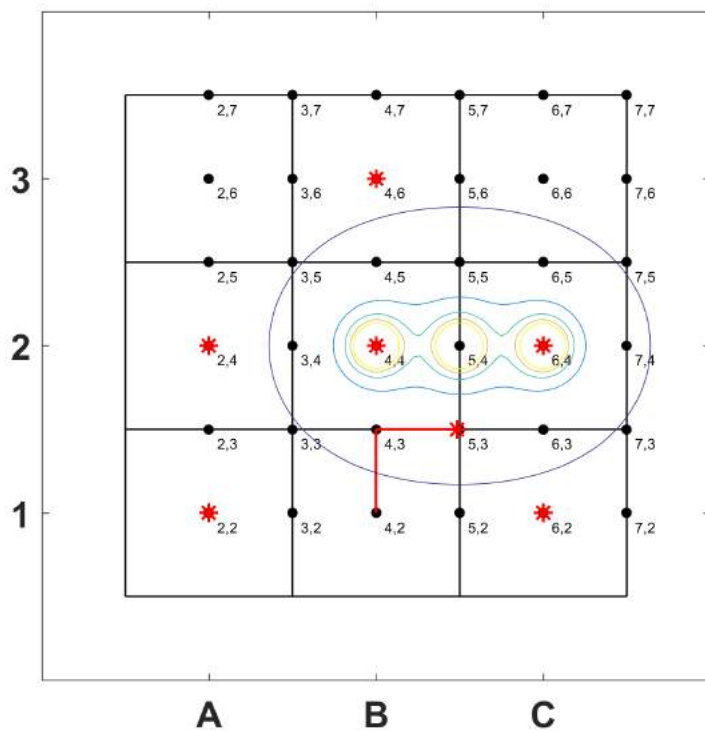
När den optimala vägen är beräknad aktiveras elektromagneterna i sekvens för att flytta pjäsen till den angivna positionen. Vid förflyttning beräknas den totala kraftvektorn från magnetfälten på samtliga pjäser. Om denna överstiger den kritiska lasten förflyttas pjäsen en kort distans i kraftvektorns riktning innan kraftvektorn uppdateras och proceduren återupprepas.

Notera att i figur 3.10 är två magneter aktiverade. Det är för att i första steget av förflyttningen när magnet (4,3) aktiveras behöver även magnet (4,4) aktiveras för att pjäsen på B2 som står över magnet (4,4) inte skall dras med i förflyttningen. Programmet fungerar genom att om en aktiv magnet angränsar horisontellt eller vertikalt till en pjäs som inte skall förflyttas aktiveras magneten under denna också.

Vid nästa steg i förflyttningen blir det tydligt varför magneterna är placerade så diagonal förflyttning inte är möjlig. Se i figur 3.11 hur pjäsen kan flyttas utan att några andra pjäser påverkas, notera hur den yttre konturlinjen av magnetfältet inte når någon annan pjäs. När denna förflyttningen är genomförd aktiveras magnet (5,4). Då behöver precis som tidigare elektromagneterna under angränsande pjäser också aktiveras för att hålla dessa på plats, se figur 3.12. I denna situation aktiveras alltså tre magneter samtidigt men trots detta påverkas inte pjäserna som är utanför rutorna B2 eller C2, se den yttre konturlinjen.



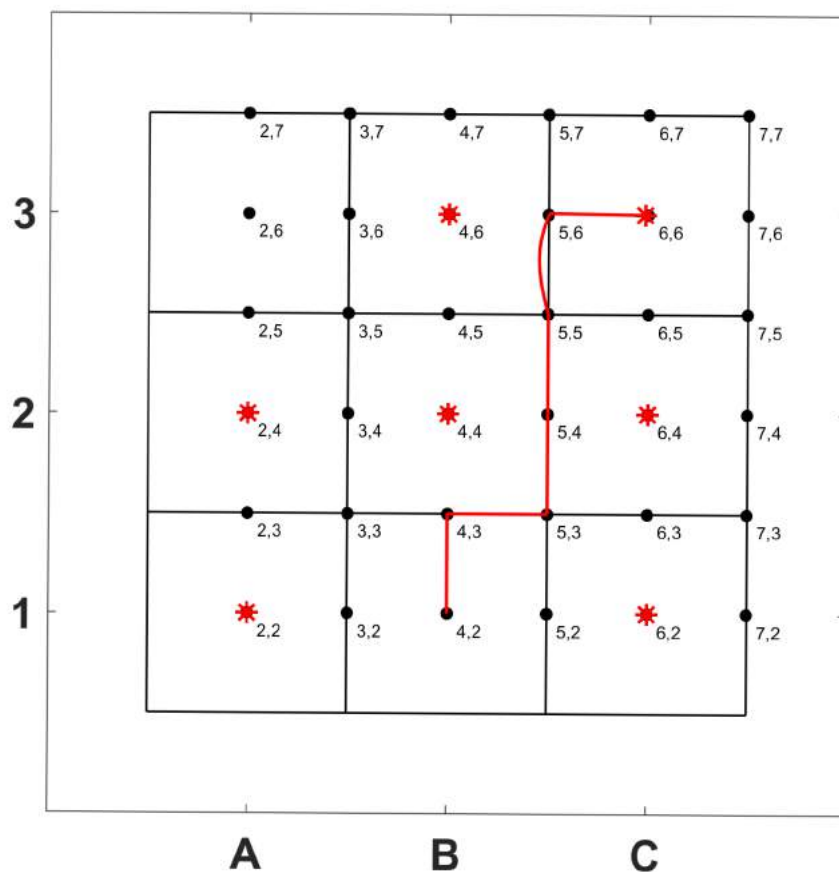
Figur 3.11: *MATLAB-simulering steg två.*



Figur 3.12: *MATLAB-simulering steg tre.*

Förflyttningen fortskrider på samma sätt tills pjäsen rört sig från startpositionen B1 till slutdestinationen C3, se hela förflyttningen i figur 3.13. Notera utböjningen vid förflyttningen mellan magnet (5,5) och (5,6). Denna beror på att när (5,6) aktiveras behöver även (4,6) aktiveras eftersom det är en annan pjäs där. Kraftfälten blir då större åt ena sidan men då (5,6) är närmare den pjäs som ska flyttas blir den resulterande positionen tillslut ändå (5,6). Det är möjligt att aktivera (6,6) samtidigt för att få en rakare rörelse, men eftersom pjäsen når sin slutliga position ändå ignoreras denna böjning.

Simuleringen har visat att det är möjligt med 9 magneter per ruta utan att behöva flytta andra pjäser än den som ska flyttas, så länge det går att använda elektromagneter som är tillräckligt lika samt placerade på rätt avstånd i förhållande till varandra. Det vill säga så att horisontell och vertikal förflyttning är möjlig men att avståndet är för långt för diagonal förflyttning. Slutsatsen från simuleringen är därför att 9 magneterskonfigurationen, 3.7, fungerar i teorin. För att dimensionera elektromagneterna och se hur lika varandra i styrka de går att göra behöver tester utföras då det är mycket svårt att dimensionera via beräkningar.



Figur 3.13: MATLAB-simulering steg sju.

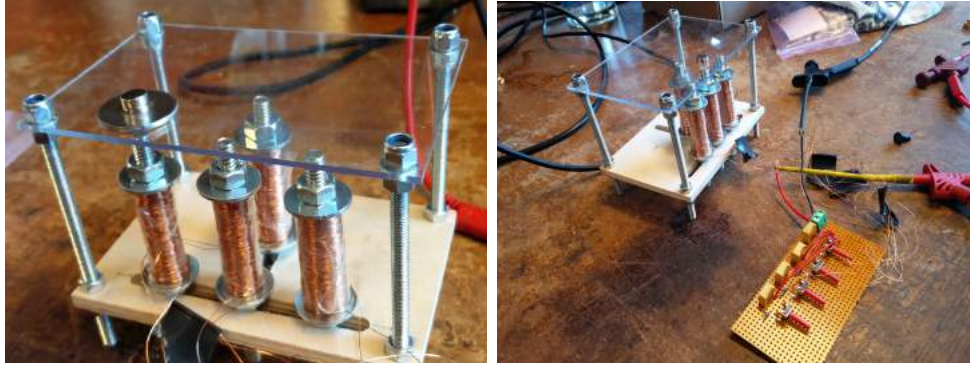
3.5 Styrka på magneter och storlek på brädet

Eftersom de variabler som påverkar styrkan hos elektromagneterna är många och innefattar komplicerade beräkningar krävs numerisk simulering av magnetfält eller tester för att få fram den styrka hos elektromagneterna som krävs i projektet. Exempelvis dras pjäsernas magneter till elektromagneternas järnkärna väldigt starkt upp till en viss höjd, och samtidigt räcker endast elektromagneternas magnetfält upp till en viss höjd, därmed behöver höjden mellan elektromagneterna och brädet dimensioneras till ett mellanläge där pjäserna dras tillräckligt löst till elektromagneternas kärnor för att kunna förflyttas. Samtidigt som detta tillstånd är uppfyllt behöver också magnetfältet spridas så pass brett att det lyckas attrahera pjäser från angränsande elektromagneters position.

3.5.1 Testrigg

För att kunna testa ifall lösningen med nio elektromagneter i varje ruta inte bara fungerade i teorin utan även i verkligheten, samt för att få praktisk erfarenhet av magneter byggdes en testrigg. Först och främst skulle det prövas ifall det gick att få pjäser att röra sig över huvud taget, sedan även om det gick att dimensionera så att rörelserna fungerade horisontellt och vertikalt samtidigt som det inte fungerade diagonalt. Sist men inte minst även om det gick att flytta en pjäs mellan två andra pjäser, vilket är den svåraste rörelse som en pjäs kan göra i och med detta konceptet. Förutom dessa tre tester som kunde bekräfta ifall konceptet fungerade var det viktigt att se hur elektromagneterna skulle bete sig för att kunna fortsätta med dimensioneringsarbetet. Saker som iaktogs var bland annat hur snabb var pjäsrörelsen var, hur stora avstånd elektromagneterna kunde dra en pjäs ifrån och hur mycket ström krävdes för att driva dem.

Testriggen består av en bottenplatta med fyra monterade elektromagneter som lindats av projektgruppen själva och en skiva 2 mm tjockt plexiglas på toppen som ska representera schackbrädet. Elektromagneterna är fastsatta i skårar så att det går att justera avståndet mellan elektromagneterna för att anpassa längden och på så sätt också styrkan. För denna skull kunde även höjden mellan plexiglas och elektromagneterna också justeras. Elektromagneterna var kopplade till ett kretskort med strömbrytare så att man kunde slå av eller på elektromagneterna manuellt. Kretskortet var sedan kopplat till ett nätaggregat som kunde variera ström och spänning efter behov. Som pjäs på ovansidan av testriggen användes två olika sorters små permanentmagneter, starka neodymmagneter och mindre starka ferritmagneter, samt vanliga omagnetiserade järnföremål. I figur 3.14a på nästa sida syns en bild på testriggen, där pjäsen består av två små neodymmagneter med en stålbricka runt sig och i figur 3.14b syns testriggen med kopplingen till kretskortet.



(a) I bilden syns testrigger med fy- (b) Här syns testrigger med kopplings- egenlindade elektromagneter fästa en till kretskortet. De små röda pin- i skenor, med en skiva plexiglas som narna på kretskortet är strömbrytare. bräde och två permanentmagneter med en bricka runt sig som pjäs.

Figur 3.14: Bilder på testrigger.

När testrigger testades så bekräftades det att konceptet fungerade som det var tänkt, det vill säga att det gick att dra pjäser horisontellt och vertikalt (sett ovanifrån) samtidigt som det inte gick diagonalt. Det gick också att det gick att flytta en pjäs mellan två andra pjäser, men att det var väldigt viktigt att magneterna var lika starka. Då elektromagneterna till testrigger tillverkades av projektgruppen hade de lindats med olika antal varv och därmed blivit olika starka. Slutsatsen som drogs var att det antingen måste gå att reglera strömmen till varje magnet för att därmed manipulera styrkan eller att använda elektromagneter med exakt samma styrka från produktion.

Elektromagneterna som projektgruppen hade gjort själva hade en kärna som var en $\varnothing 5$ mm stålstav med gängade kanter. På varje kant satt en mutter påskruvad med en bricka på insidan. Mellan brickorna blev det ett 50 mm stort gap som fylldes med ca 1600 varv $\varnothing 0.25$ mm koppartråd. Tråden rullades på stålstaven genom att snurra den med hjälp av en skruvdragare. Detta var tidsödande och det var svårt att hålla räkningen så att alla fick exakt samma antal varv och därav samma styrka. Genom att använda en svarv som håller räkning på antalet rotationer istället för en skruvdragare hade elektromagneterna kunnat bli mer lika i storlek och styrka. Detta ansågs dock både besvärligt och tidsödande, varvid projektgruppen vände fokus från att tillverka egna elektromagneter till att köpa in dem.

Att hitta färdiga elektromagneter som var tillräckligt billiga och stora var svårt. De billigaste som erbjöds kostade 3.5\$ styck, vilket med ett inköp på 285stycken skulle innebära ett inköp högt över projektets budget. Dessa hade likt de flesta andra liknande elektromagneter en storlek i största laget för schacket. Vid konsultation med Robert Karlsson vid institutionen för Elektroteknik tipsade han om att det går att använda en elkompnent kallad drossel som en elektromagnet. Dessa är ofta mycket små och billiga komponenter menat för ett annat syfte än som elektromagneter. Ett antal drosslar av olika storlekar och sorter beställdes därför in för att kunna testas, vilket beskrivs noggrannare under nästa avsnitt.

3.5.2 Test av drosslar

Drosslar är spolar och används ofta som en passiv induktor som blockerar högfrekvent växelström i en elektrisk krets men släpper fram signaler med lägre frekvens. Trots att drosslarna inte är avsedda för att användas som elektromagneter är detta fullt möjligt. Ett antal olika drosslar beställdes hem och testades, dessa kan ses i figur 3.16. Den drossel som är gulmarkerad är den som tillslut valdes, notera att denna inte var med i första beställningen utan beställdes efter utvärdering av resultatet från testerna av tidigare beställda drosslar.

Testerna genomfördes med tre typer av möjliga pjäser, två cylindriska neodymmagneter med måtten $\varnothing 8 \times 3$ mm respektive $\varnothing 5 \times 1$ mm samt en cylindrisk ferritbit med måtten $\varnothing 8 \times 3$. Drosslarna placerades på en kopplingsplatta och kopplades till ett nätaggregat. En plexiglasskiva ovanför drosslarna justerades upp och ned med hjälp av 4 gängade stänger för att snabbt kunna testa olika höjder. Hela testuppsättningen visas i figur 3.15. Pjäserna placerades på plexiglasskivan ovanför en inaktiv drossel och testet gick ut på att flytta pjäsen till en närliggande aktiv drossel. Höjden varierades genom att sakta höja plexiglasskivan från drosslarna, först vid 300 mA för att sedan upprepas med 500 mA, 1 A och 1.5 A. Vid högre strömmar noterades ingen ökning av magnetfältets styrka. De allra flesta drosslarna föll bort redan här, då de inte lyckades med att flytta någon av de magnetiska material som testades.

Den gemensamma faktorn för de drosslar som fungerade var att de hade tunn tråd och många varv. Problemet med de mindre drosslarna är att magneterna i pjäserna hamnar nära varandra och att de då repelleras. Då magneterna tyngs av en pjäs kan de troligen vara närmre varandra än när de ligger fritt förutsatt att drosslarna klarar av att flytta även dessa. De små drosslarna var dock för svaga för att flytta magneter med tyngd.

De större drosslarna som testades hade tjockare tråd med få varv och genererade inte ett tillräckligt starkt magnetfält. För att åtgärda detta lindades tråden av från en av de större drosslarna och ersattes med den tunna koppartråd som användes till de egenlindade elektromagneterna. Denna egenlindade drossel klarade det test som hade satts upp på ett tillräckligt stort avstånd för att magneterna skulle ligga still på två inaktiva intilliggande drosslar. För den större neodymmagneten är det precis på gränsen men med den extra tyngden av en pjäs bör det inte vara några problem. Den mindre magneten fungerar bra.

För att hitta en färdig drossel med liknande specifikationer som den egenlindade mättes dess induktans och resistans. Dessa värden uppmättes till 10 mH och 4,4 ohm och användes som filter vid fortsatta sökningar. Detta resulterade i att ytterligare en drossel, Panasonic ELC18B103L, beställdes vilken fungerade mycket bättre än någon av de drosslar som tidigare testats. Den hade dessutom fördelen att den har en jämn bas i plast vilket förenklar installationen avsevärt, då alla elektromagneter kunde installeras med samma avstånd infrån kopplingsplattan. Drosseln kördes framgångsrikt med en ström på 1 A och en spänning på 5 V. Nedan finns en bild från ett test med denna drossel som sedan kom att användas till den slutgiltiga konstruktionen, se figur 3.15.



Figur 3.15: I bilden syns testuppsättningen med 4 drosslar av den typ som kommer att användas för schacket.

Testade drosslar						
Artikelnummer (tillverkaren)	Märkström	Induktans	Resistans (DC)	Diameter	Sort	Kommentar
ELC11D102F	0.57A ±10%	1mH	1,1 Ω	11,5 mm	Radiell	Drar något, men lite för dåligt för att klara rutor större än 4 cm
11PHC-473K-50	0.07A ±10%	47mH	97 Ω	12 mm	Radiell	Något bättre än ovanstående
07HCP-330K-50	1.5A ±10%	0,033 mH	0,1 Ω	8,3 mm	Radiell	Alldeles för svag, ingen skillnad mellan av och på
07HCP-332K-50	0.14A ±10%	3.3mH	9,3 Ω	8,3 mm	Radiell	Väldigt svag
07HCP-103K-50	0.14A ±10%	10mH	24 Ω	8,3 mm	Radiell	Bättre än den ovanstående, kanske duger för små pjäser
RLB1314-153KL	0.082A ±10%	15 mH	15 Ω	11,7 mm	Radiell	Knappt någon styrka, antagligen pga den mycket låga strömmen.
SDR1006-102KL	0.23A ±10%	1 mH	3,1 Ω	9,8 mm	SMD	fungerar ej, svag (Hade lite svårt att testa de ytmonterade)
SDR1006-3R3ML	5A ±20%	0,0033 mH	0,024 Ω	9,8 mm	SMD	fungerar ej (Hade lite svårt att testa de ytmonterade)
SDR1006-121KL	0.89A ±10%	0,120 mH	0,4 Ω	9,8 mm	SMD	fungerar ej (Hade lite svårt att testa de ytmonterade)
SDR1006-150ML	2.25A ±20%	0,015 mH	0,08 Ω	9,8 mm	SMD	fungerar ej (Hade lite svårt att testa de ytmonterade)
HM78D-128221KLFTR	1.6A ±20%	0,22 mH	0,345 Ω	12,5 mm	SMD	fungerar ej, skämdad? (Hade lite svårt att testa de ytmonterade)
ELC18B103L	0.36A ±10%	10mH	3,9 Ω	16 mm	Radiell	Fungerar helt klart bäst i test.

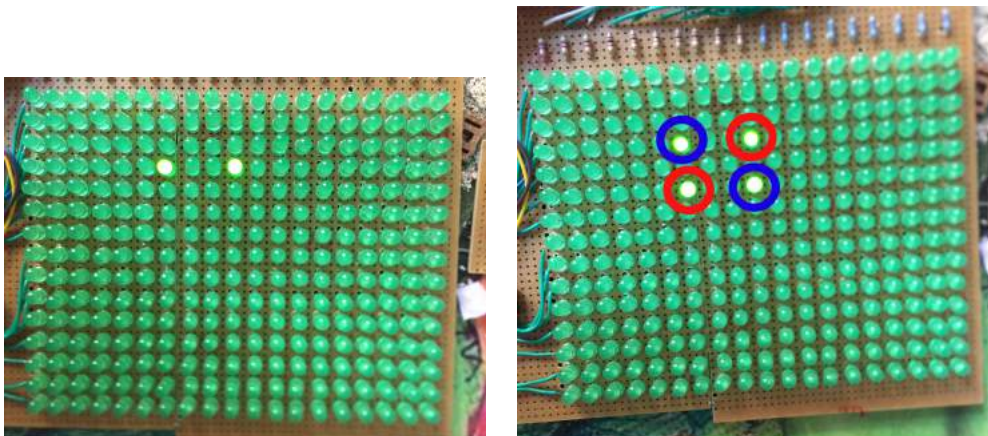
Figur 3.16: I tabellen presenteras de drosslar som testades, den med gul bakgrund är den drossel som användes för den slutgiltiga konstruktionen.

3.6 Ledmatris

En matris av LED-dioder konstruerades för att kunna jobba vidare med mjukvaran parallellt med att hårdvaran konstruerades. LED-matrisen bestod av 285 stycken LED dioder placerade i ett rutnät med 15 rader och 19 kolumner, det vill säga samma mönster som magneterna skulle ligga i. Simuleringar av denna lösning är möjliga, men då de elektriska kopplingarna kommer bli likartade mellan LED matrisen och den färdiga hårdvaran gav bygget av LED-matrisen viktig erfarenhet i hur det kommer se ut med kablage och komponenter i hårdvaran.

Det viktigaste kravet med matrisen var att dioderna skulle kunna styras individuellt och helt oberoende av varandra, för annars skulle det resultera i att pjäsernas rörelser i den färdiga produkten inte kommer vara möjliga. Lösningen för LED-matrisen bygger på att man seriekopplar anoderna (positiv pol) av dioderna i kolumnerna och katoderna (negativ pol) av dioderna i raderna, detta möjliggör att dioderna kan tändas individuellt genom att lägga på spänning på dess rad resp kolumn. Ingångarna från varje rad respektive kolumn kopplades in i shiftregister, vilket möjliggör styrning med färre ingångar än att koppla varje enskild magnet till arduino-kortet, detta beskrivs mer ingående i kapitel 4.

Testning av matrisen klargjorde att styrning av en diod i taget var möjlig, samt att styrning av flera dioder på samma rad eller kolumn samtidigt var möjligt, vilket kan ses i figur 3.17a. Vid försök att styra flera samtidigt dioder som inte var på samma rad, upptäcktes det att oönskade dioder aktiverades, vilket kan ses i figur 3.17b. Detta gjorde att denna typ av matris inte kan användas om man vill kunna flytta på fler pjäser samtidigt, som till exempel när en pjäs slås ut. Vid förflyttning av en pjäs i taget fungerar konceptet eftersom flera magneter endast behöver aktiveras när en pjäs passerar en annan och att de då ändå är på samma rad eller kolumn. Det finns ett fåtal specialfall där magneter på olika rader och kolumner behöver aktiveras samtidigt gällande hantering och placering av döda pjäser, men dessa kan kringgåas genom ändringar i koden.



(a) Två dioder på samma rad är tända.

(b) Dioderna med blå ring runt är tända och som ett följdfel blir även dioderna med röd ring tända.

Figur 3.17: Ledmatrisen i två olika lägen.

För att kringgå problemet och möjliggöra förflyttning av flera pjäser samtidigt krävs det att alla elektromagneter kan sättas på och stängas av helt individuellt. Detta uppnås med hjälp av att istället för att ha rader och kolumner inkopplade i shiftregistrerna, ha fler shiftregister och koppla varje elektromagnet till en enskild utgång. Mer om shiftregister går att läsa om i komponentavsnittet 4.1.1. Den koppling som kommer användas beskrivs under avsnitt 4.1, Elektrisk koppling.

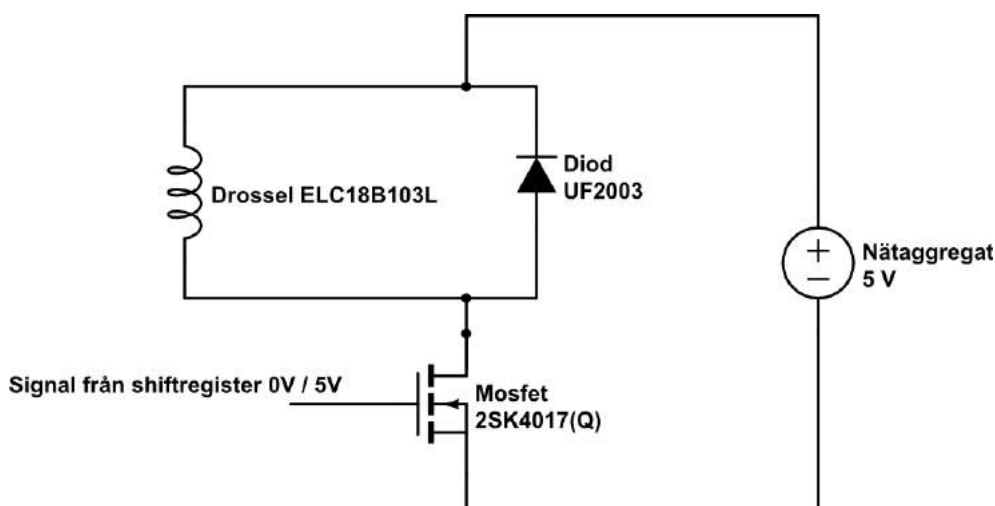
På grund utav denna motgång tillförde inte ledmatris mycket till själva testandet av programmerings logiken som var dess ursprungliga mål, dock hjälpte det avsevärt till projektet som helhet då planen från början var att koppla elektromagneterna enligt samma koppling. Att hitta felet tidigt innan beställning av komponenter och kopplingsplattor ens skett medförde både sparande i tid och pengar.

4. Konstruktion av prototyp

I föregående kapitel behandlades vidareutvecklingen av konceptet. Detta kapitel är till för att beskriva prototypens konstruktion. För att allting i schackbrädet ska kunna samverka finns det flertalet dellösningar som behöver fungera, kretsen behöver strömförsörjning och behöver kunna styras, det behövs mjukvara som sköter dragen och ser till att förflyttningar sker på önskvärt sätt. Slutligen ska allt kopplas ihop med den konstruktion som utgör det fysiska schackbrädet.

4.1 Elektrisk koppling

I detta avsnitt beskrivs och förklaras den elektriska kopplingen av komponenterna. Varje komponent listas och förklaras under avsnitt 4.1.1, Komponenter. Uppkopplingen för en elektromagnet ser ut som i bilden i figur 4.1. Till ena änden på spolen är matningsspänning, +5 V inkopplat. Eftersom elektromagneterna är spolar är de strömtröga, därför måste man se till att det inte finns någon möjlighet för den kvarvarande strömmen att ta vägen någonstans då kretsen bryts eller då ström kan induceras i spolen. Detta har lösts genom att koppla en diod parallellt över spolen, vilket alltså medför att ström inte kan röra sig åt fel håll i kretsen. Spolens andra ben är kopplat till Drain på transistorn. Transistorns source går till jord medan transistor-gaten är kopplad till en utgång på ett så kallat shiftregister vilket förklaras i stycke 4.5.

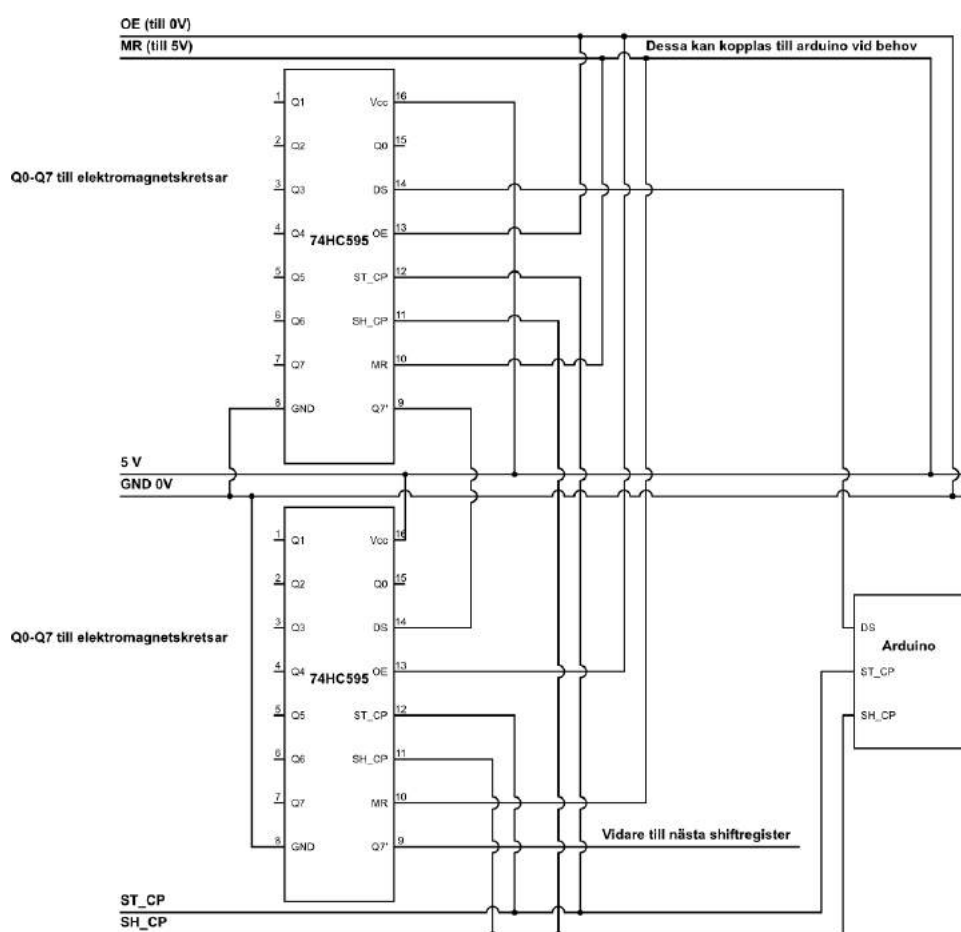


Figur 4.1: Kopplingsschema för en elektromagnet.

För att kunna styra alla 285 elektromagneter behöver de kunna stängas av och sättas på individuellt. Vilket innebär att varje elektromagnet skall sättas på var sin utgång från mikrocontroller-kortet, men eftersom det inte finns färdiga kort med tillräckligt många utgångar används så kallade shiftregister kopplade enligt figur 4.2. Dessa shiftregister möjliggör att med relativt få utgångar från mikrocontroller-kortet hantera en stor mängd signaler. Vidare skickas shiftregisterns signal till varje elektromagnet, men eftersom elektromagneterna drivs med en betydligt högre ström än vad resterande komponenter

i kopplingen gör används den ovannämnda transistorlösningen 4.1. För att sköta logiken bakom rörelserna (alltså vilka elektromagneter som ska sättas på och av) och fylla i shiftregisterna används ett mikrokontroller-kort, mer specifikt ett Arduino-kort eftersom det är både lätthanterligt och smidigt, men främst eftersom projektgruppen haft tidigare erfarenhet med kortet.

Sammanfattningsvis går en signal som skall sätta på en elektromagnet från ett Arduino-kort till ett shiftregister som slutligen aktiverar en transistor som därmed fungerar som en knapp och sätter på elektromagneten. För att driva allt i kopplingen används två spänningskällor, en som kan hantera höga strömmar som driver elektromagneterna, samt Arduino-kortets spänningskälla (via USB till en dator) som driver Arduino-kortet och shiftregister.



Figur 4.2: Kopplingschema för ihopkopplade shiftregister.

För att elektromagneterna ska dra till sig pjäser med tillräcklig kraft krävs mycket ström. I databladet till de utvalda drosslarna kan man se vid vilken ström som drosseln mätts [20]. Är en elektromagnet mättad betyder det att den har nått toppen av den exponentiella kurva som förklarar styrkan på det skapade magnetfältet med avseende på ström. Detta innebär i sin tur att mer tillförd ström inte kommer att påverka magnetfältet lika kraftigt som tidigare, en rimlig mängd ström som lyckades flytta pjäser vid test var 1Ampere.

Vidare mättes drosslarnas interna resistans upp till 4.4Ω vilket tillsammans med den minimala resistans i resten av kretsen (i form av kablar och kopplingskenor) skulle innebära att 5V skulle vara en lämplig drivspänning enligt Ohms-lag ($U=RI$). För att kunna driva flera elektromagneter på en och samma gång som var och en drar 1A skulle en spänningskälla som klarar av att leverera så många ampere som möjligt vara att föredra.

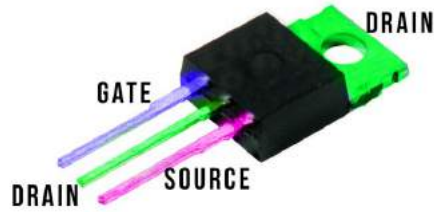
4.1.1 Komponenter

- **Elektromagneter** Den typ av elektromagnet som valdes var Panasonic ELC18B103L. Mer information om valet går att läsa under avsnitt 3.5.2 Test av drosslar. Kortfattat var det den produkt bland de som testats och undersökts som bäst klarade av förflyttning och samtidigt låg inom budget.
- **Dioder** En diod är en elektrisk ledare som idealt endast leder ström i en riktning. Eftersom elektromagneterna är spolar kan spänningen över spolen beräknas enligt $V = L \frac{di}{dt}$ där L är induktansen. Notera att då strömmen stängs av blir $\frac{di}{dt}$ -termen kraftigt negativ vilket ger en negativ spänningsspik över spolen och därmed en mycket stor spänningsskillnad över transistorn. Detta fenomen kallas strömtröghet, det vill säga att elektronerna vill fortsätta vandra trots att spänningskällan stängts av.

För att undvika att transistorn skadas parallellkopplas en diod med spolen på ett sådant sätt att strömmen endast kan vandra tillbaka då potentialen över spolen är negativ, se kopplingsschemat i figur 4.1. Spolen kan då dra elektroner från sig själv tills dess att spänningsskillnaden dämpats till följd av resistans. De dioder som används är MULTICOMP UF2003 på grund utav att dess specifikationer inte skulle överskridas inom kretsen samt dess pris.

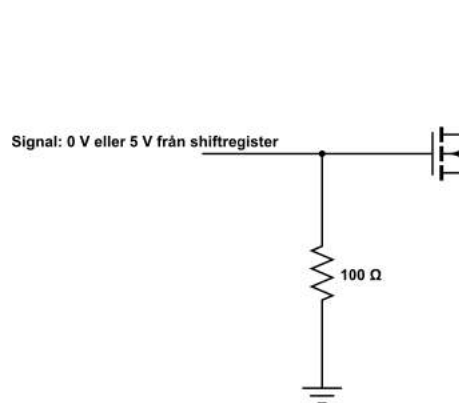
- **Transistorer MOSFET** För att generera ett tillräckligt starkt magnetfält krävs att ungefär 1 ampere leds genom elektromagneterna. Ett Arduino-kort klarar endast att leverera 20-40 mA vilket innebär att en annan strömkälla är nödvändig. Varje elektromagnet kopplas till en transistor som används som en strömbrytare vilken i sin tur kan styras av Arduino-kortet.

I detta projekt används transistorer av typen Mosfet. Den stora skillnaden mot en BJT, en annan vanlig transistortyp, är att endast en spänningsskillnad och ingen ström krävs för att aktivera komponenten. En Mosfet har tre ben, Gate, Source och Drain och kan vara av typen N-kanal eller P-kanal. Benet av typen Gate fungerar som en strömbrytare för strömmen mellan de två andra benen, från Drain till Source. För att aktivera komponenten krävs en positiv spänningsskillnad från Gate till Source. En Mosfet av typen N-kanal leder endast ström då det finns en spänningsskillnad, en Mosfet av typen P-kanal leder endast ström då det inte finns en spänningsskillnad. En bild på en Mosfet-transistor syns i figur 4.3 [21].



Figur 4.3: Mosfet med de tre benen Gate, Drain och Source.

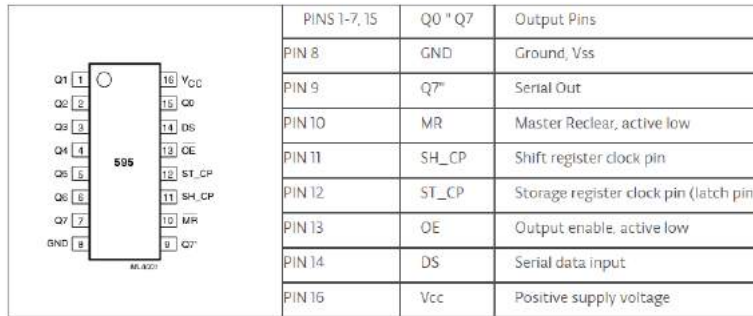
Eftersom endast en spänningsskillnad och ingen ström krävs för att aktivera komponenten innebär det att ingen seriekopplad resistor innan Gate är nödvändig för att reglera strömmen. Ett vanligt sätt att koppla från signal in till Gate på en N-kanals Mosfet är att också koppla ett motstånd till jord, detta görs för att man ska vara säker på att det verkligen blir 0V på Gate då inte 5V ges. Det shiftregistret som används matar ut max 0,1 V då benet är satt till lågt vilket innebär att ett sådant motstånd inte är nödvändigt. För att tydligare visa detta visas motståndet i figur 4.4, som alltså inte behövs i detta fallet.



Figur 4.4: MOSFET-transistor med resistor mellan jord och gaten, vilket alltså inte används.

De krav som ställs på MOSFET-transistorn är att den ska klara 1 A vid 5 V. Resistansen hos elektromagneterna är cirka 4,4 ohm, d.v.s för att komma upp i 1 A vid 5 V måste resistansen hos transistorn vara under 0,6 ohm enligt $U = RI$. Transistorn behöver också ha en tröskelspänning på under 5 V för att kunna aktiveras av shiftregistret. Den transistor som används för detta projekt blev därmed TOSHIBA 2SK4017(Q). Det är en MOSFET av typen N-Kanal som klarar 5 A och 60 V. Den har inre resistans på 100 mohm och en tröskelspänning på 2.5 V, vilket därmed uppfyller de satta kraven på komponenten.

- **Shiftregister** Eftersom det inte existerar 285 stycken utgångar på ett Arduino-kort används shiftregister. Dessa komponenter möjliggör att göra om parallell kommunikation till seriell och att man därmed kan, i teorin, omvandla projektets 285 utgångar till bara tre. Nedan i figur 4.5 visas de ingångar och utgångar som finns tillgängliga hos shiftregister 74HC595 [22] som används i projektet.



Figur 4.5: Märkning och beskrivning av in-och utgångar till shiftregister 74HC595.

Ett shiftregister fungerar som ett register som fylls på uppifrån. Succesivt läggs värden in i registret en bit i taget via den så kallade *serial data input pin* (seriell in port) och klockar in dem i registret med hjälp av *shift register clock pin* (klockpinne). Detta innebär alltså att när klockpinnen aktiveras läggs värdet från den seriella inporten längst upp i registret och skjuter ner alla föregående värden. Med andra ord: om registret innehåller 8 bitar behövs 8 stycken klockningar för att helt skriva över de föregående värdena. När nya värden ska läggas in i registren inleds processen med att inaktivera den så kallade *latch pin*, som ser till att inga ändringar tar sig till output under tiden nya värden klockas in. När så småningom alla klockningar har gjorts och det önskade värdet finns i registret aktiveras *latch pin* som då skickar ut värdet i registret till dess utgångar.

Sammanfattningvis förklaras några av benen, vars funktion inte är helt intuitiv [22]:

- **Serial data input pin:** Benet där data som ska läggas in i registret placeras.
- **Storage register clock pin (latch pin):** Benet som aktiveras för att lägga ut värdet från registret till output.
- **(Shift register clock pin):** Benet som aktiveras för att lägga in värdet från data input längst upp i registret.
- **Master Reclear, active low :** Benet ges 0 V för att nollställa alla utgångar, behövs inte i detta projektet del utan den kopplas till 5 V. Kan styras från Arduino-kortet om önskvärt.
- **Output enable, active low :** Möjliggör att de parallella utgångarna kan användas, kopplas därför till 0 V. Även denna kan styras från Arduino-kortet om önskvärt.

I projektet används som sagt shiftregister av typen 74HC595, dessa har totalt 8 stycken utgångar per komponent, vilket innebär att endast 8 elektromagneter kan styras per register. Detta innebär att det behövs: $\frac{285}{8} = 35.625 \approx 36$ stycken shiftregister för att styra hela kopplingen. I och med att shiftregistren kan kopplas ihop genom att använda det föregående registrets serial out pin som data input för nästa register kan totalt 3 utgångar användas från Arduino-kortet för att styra samtliga elektromagneter.

- **Arduino** Ett Arduino-kort är ett så kallat mikrocontroller-kort, vilket enbart utgör en del av en normal dator [23]. Kortet kör inte ett operativsystem utan exekverar enbart den kod som laddas in på kortet. Detta medför att många funktioner från ett operativsystem inte finns, men direkt exekvering av simpel kod blir både lättare och

snabbare. Det huvudsakliga syftet med ett Arduino-kort är att utgöra ett gränssnitt mellan sensorer och andra enheter i ett system. Därför är ett Arduino-kort typiskt bra när ett system ska svara på olika sensorer eller inmatning.

I fallet för det här projektet som ska styra en hel uppsättning elektromagneter krävs det både en dator för att kunna implementera schackalgoritmen och ett gränssnitt för ett stort antal utgångar för alla 285-elektromagneter. Därför används en Arduino UNO som utgör gränssnittet mellan elektromagneterna och logiken, samt preliminärt en dator kopplad till Arduino-kortet. Att just modellen Arduino UNO används är för att projektgruppen har tidigare erfarenheter med den och den har funktioner som passar för ändamålet. Datorn sköter i sin tur schacklogiken och ger kontinuerlig input till Arduino-kortet i form av rörelser på brädet. Detta medför därmed möjligheter för framtida utveckling att implementera en Raspberry Pi (som är ett individuellt kort mer likt en dator) som kör schacklogiken och därmed separera hela schackuppsättningen till en självständig produkt.

- **Nätaggreat** De krav på nätaggreat som tidigare nämnts var en spänning på 5V och tillräckligt med strömförsörjning för att driva ett antal elektromagneter samtidigt då de drar cirka 1 Ampere styck. Ett gammalt nätaggreat från en dator visade sig vara lämpligt då det kunde leverera 5V och 15A. Dock saknade detta nätaggreat ett överbelastningskydd vilket ledde till inköp av säkringar, för att undvika att nätaggreatet bränns. Om man skulle använda nätaggreatet för att driva många elektromagneter samtidigt kan det bli problem med kopparbanorna på kretskorten eftersom de är väldigt tunna och kanske inte tål så pass mycket ström. Vilket skulle kunna innebära att man behöver flytta över hela kretsen till ett mer strömtåligt kretskort, eller alternativt möjligen modifiera det tillgängliga kretskortet.

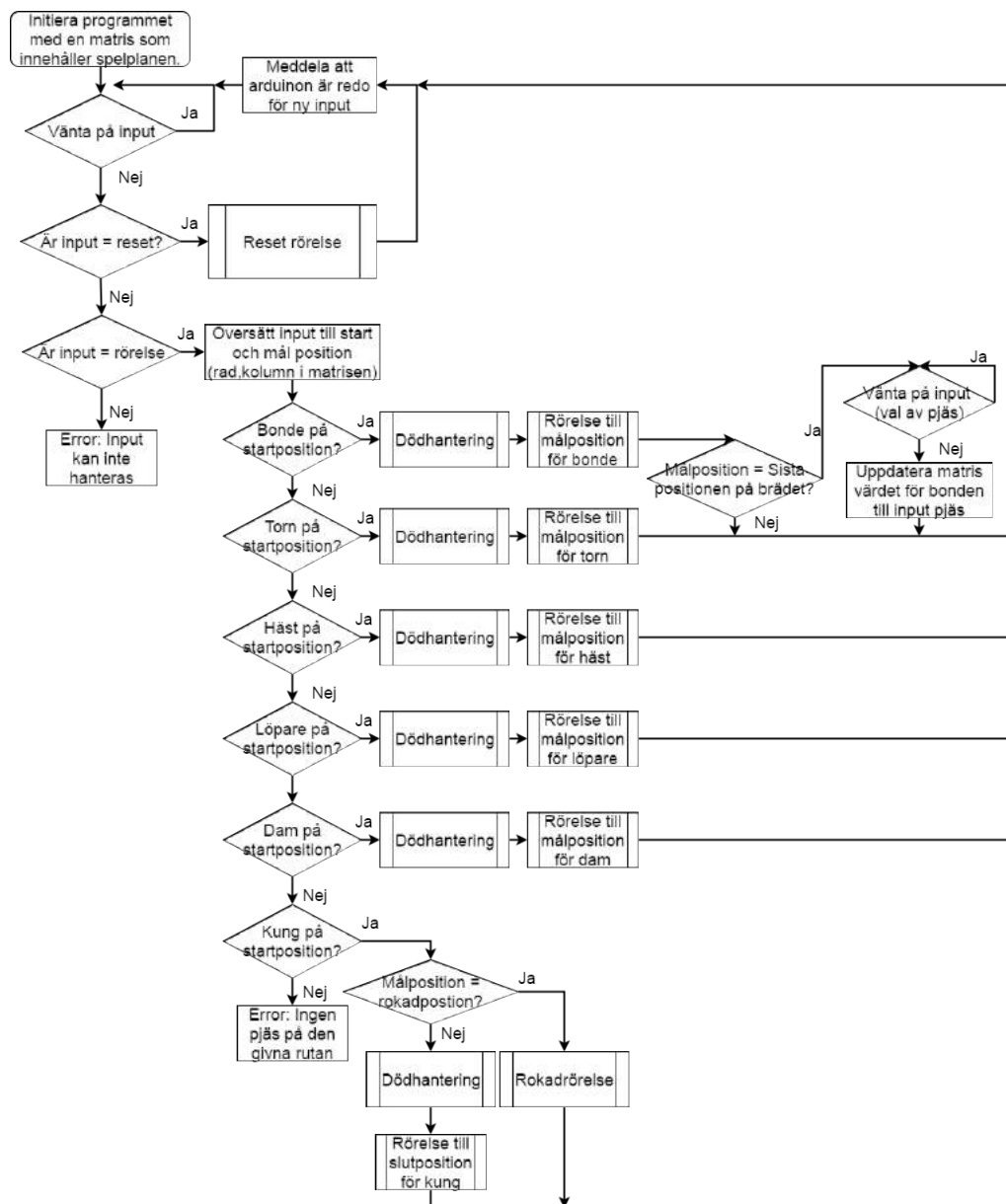
4.2 Mjukvara

Förutom hårdvara krävs logik för att styra schackbrädet. Först och främst krävs logiken för Arduino-kortet, som i detta projekt gjorts från grunden och sköter själva rörelsen igenom att sätta på rätt elektromagneter i sekvens. Samtidigt krävs ett användargränssnitt som utgör kopplingen mellan användare och dator samt vidare till Arduino-kortet, som i detta projekt baseras på en open source schackmotor med tillhörande användargränssnitt. Denna schackmotor är skriven i Java av Alberto Alonso Ruibal samt Lukas Laag [24] och kallas *Carballo*, dessutom sköter det tillhörande användargränssnittet själva schackspelen och spelarnas inmatning. Nedan förklaras de bägge delarna av mjukvaran i form av flödesscheman och förklaringar av viktiga funktioner.

4.2.1 Arduino firmware

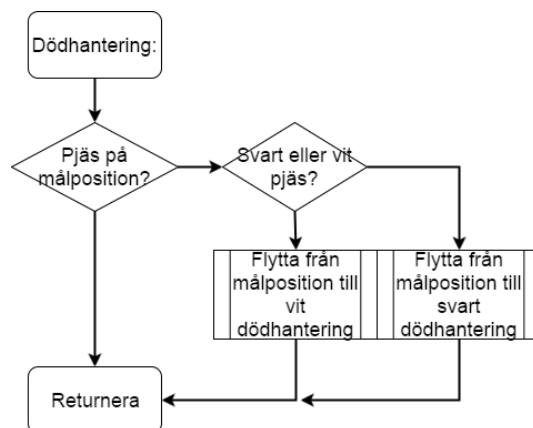
För att få en helhetsbild av programmet presenteras nedan flödesscheman för olika delar och funktioner ur programmet. Först och främst presenteras ett förenklat schema över hela programmet, som visar den konstanta loop som Arduino-kortet befinner sig i. Programmet initieras med en stor matris med 15x19 platser som skall motsvara brädet, denna matris initieras till att innehålla pjäserna i sin startposition som sedan kommer att manipuleras i någon av de många rörelsefunktionerna. Anledningen till att denna matris behöver användas och initieras är på grund utav att ingen sensoravkänning om var pjäser befinner sig vid en given tidpunkt implementerats, utan programmeringen grundas i en normal startuppställning som sedan manipuleras. Att implementera sensoravkänning skulle göra det möjligt att både verifiera att en rörelse har lyckats och att återställa ett bräde vid ny uppstart av programmet.

Vidare så väntar programmet efter sin initiering på input från sin seriella ingång(USB). När en input har tagits emot förväntar sig Arduino-kortet antingen en rörelse eller ett återställnings kommando (återställning kommer hädanefter skrivas som reset) och om inget av det tas emot skickas ett felmeddelande till användaren. Om ett resetkommando tagits emot (nytt spel i användargränssnittet) loopar programmet igenom sin nuvarande matris och utför rörelser som flyttar tillbaks alla pjäser till sina respektive startpositioner. Eftersom startpositionen hos en given pjäs kan vara upptagen, utförs hemrörelsen rekursivt tills dess att en tom plats har hittats eller tills funktionen fastnat (att slutpositionen hos någon av pjäserna är upptagen av en annan pjäs i kedjan) då en godtycklig temporär rörelse behöver göras. Om input istället är en rörelse, vilket är definierat i koden som en sträng med startposition och slutposition enligt "a3b3" där startpositionen är a3 och slutpositionen är b3 på brädet, översätts dessa positioner till dess respektive rad-och kolumn-position i den definierade matrisen. Därefter kontrolleras vilken pjäs som befinner sig på startpositionen i matrisen, därefter kollar programmet om det finns en pjäs på slutpositionen som ska slås ut, detta görs med dödhanteringsfunktionen som förklaras kort i nästkommande flödesschema, 4.6. Beroende på vilken pjäs som befann sig på startpositionen rör sig pjäserna i olika kombinationer av raka rörelser, vilket kommer att förklaras i efterkommande flödesschema, 4.9. Slutligen när dessa rörelser har utförts meddelar Arduino-kortet på dess seriella utgång (USB) att den är redo för ny input.



Figur 4.6: Förenklat flödesschema över Arduino-programmets kod.

För att hantera döda (utslagna) pjäser har det som tidigare nämnts i slutet av avsnitt 3.3 lagts till två kolumner av elektromagneter på var sin sida av det ursprungliga brädet på 8x8 rutor som motsvaras av 15x15 elektromagneter. Dessa extra kolumner är tänkta att vara platser dit utslagna pjäser flyttar sig. De specifika platser som de döda pjäserna placeras på går att se som dödsfall på den slutgiltiga designen av brädet som kan ses i delen om konstruktion av brädet 4.3, figur 4.15. Vidare fungerar dödhanterings-funktionen så att den först kontrollerar om slutpositionen från den inkomna rörelsen innehåller en pjäs. Om detta inte är fallet sker ingenting och startpjäsen kan börja sin rörelse till slutpositionen. Om slutpositionen har en pjäs på sig betyder detta att denne pjäs skall slås ut och då beror färgen på pjäsen roll till vilken sida den utslagna pjäsen skall förflytta sig. När sidan är bestämd behöver den även hitta rätt position, vilket förklaras nedan och sedan förflytta sig dit.



Figur 4.7: Förenklat flödesschema över hur dödhanteringen fungerar.

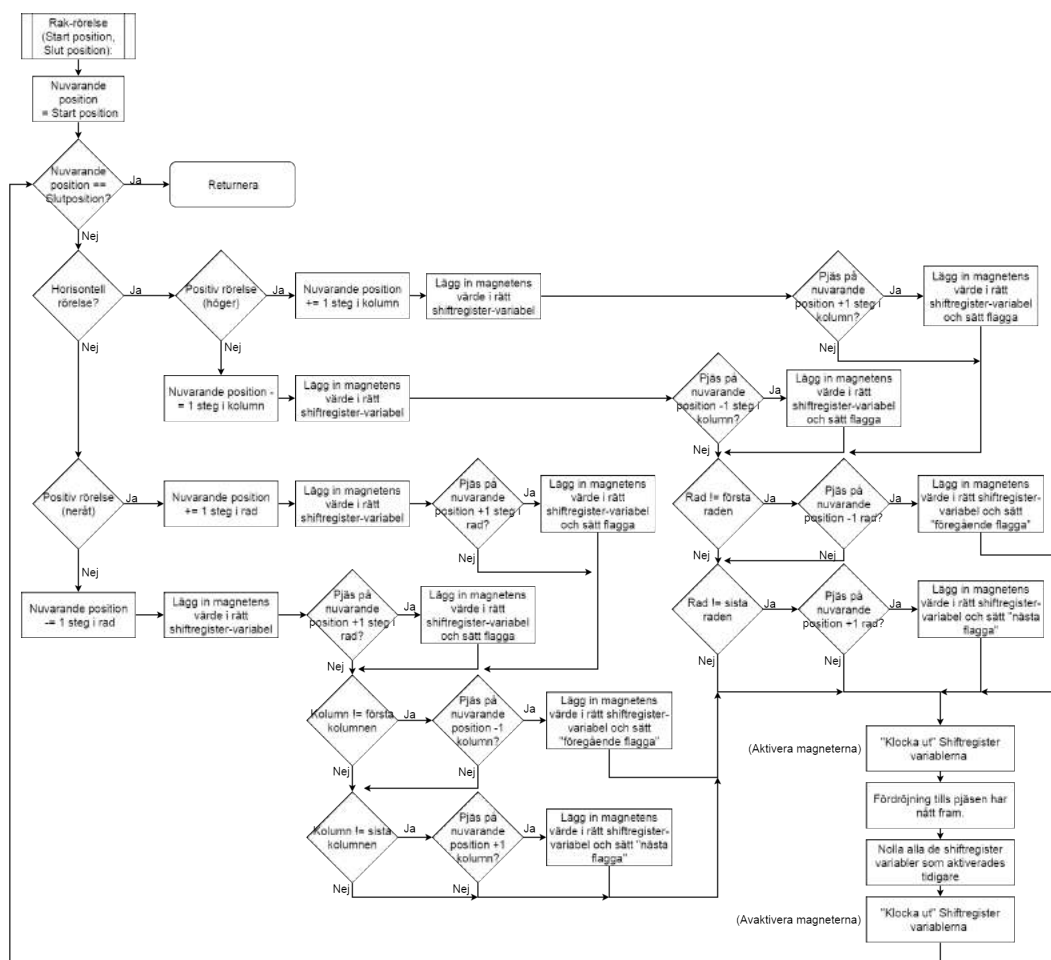
Hur extrakolumnerna fylls i är relativt komplicerat och anledningen till detta är att försöka ha så få magneter igång samtidigt och att inte behöva aktivera magneter som är mer än ett steg bort från den pjäs som flyttas. De platser som fylls i dödhanterings-kolumnerna först är varannan plats i den yttersta kolumnen upp tills att 8 pjäser slagits ut, därefter fylls varannan av de kvarvarande platserna i den yttersta kolumnen vilket medför att det maximalt sätts på 3 magneter samtidigt i någon av dessa rörelser. Sist, om det är så att mer än tolv pjäser slås ut för en färg i ett givet parti kommer tre platser i den näst yttersta kolumnen som är på samma rad som de tre tomma platserna i den yttersta kolumnen fyllas. Hur pjäserna placeras och i vilken ordning visualiseras också i figur 4.8. Därmed skulle en full dödhanterings-yta innehålla 15 pjäser, kungen exkluderad eftersom spelet är slut om kungen slås ut, varav 12 pjäser befinner sig i den yttersta kolumnen i grupper om 3 och de resterande 3 pjäserna som befinner sig i den näst yttersta kolumnen på var fjärde plats. Dessutom fyller en given sida (färg) dödhanteringen till höger med motståndarens pjäser uppifrån och ner, vilket medför att denna påfyllning av utslagna pjäser blir spegelvända sinsemellan beroende på färg.



Figur 4.8: Bilden visar hur pjäserna placeras i dödhanteringen.

Nedan följer flödesschemat med förklaring över en rak rörelse mellan två positioner på antingen samma rad eller samma kolumn, figur 4.9. På grund utav att antalet magneter per ruta som valdes att användas i projektet är 9 definierades ett problem angående att pjäserna inte får röra sig diagonalt emellan elektromagneterna, då det skulle medföra att en pjäs inte kan röra sig i mitten emellan rutor som nämndes i 3.3. Detta innebär därmed att eftersom elektromagneterna är dimensionerade till att endast dra en pjäs horisontellt eller vertikalt, att alla olika rörelser inom schacket är uppbyggda av olika kombinationer av raka rörelser.

Anledningen till att det blir olika kombinationer är att vissa rörelser skall ske emellan rutor som exempelvis hästens eller rockadens samtidigt som andra rörelser som bonden eller damen skall ske rakt igenom rutor. Dessutom så finns damen och löparen som skall röra sig diagonalt på brädet, denna rörelse löses genom att gå vartannat rakt steg lodrätt och vartannat rakt steg horisontellt i rätt riktning. Exempelvis skulle en hästs rörelse som skall tre rutor upp och en ruta till höger, ta ett steg åt höger till mellankolumnens för att sedan göra en rak rörelse upp till målraden och sist röra sig ett steg åt höger för att nå sin målposition när det gäller elektromagneterna. Att nämna är att Arduinons förflyttningar inte använder sig av A* som användes för simuleringen av konceptet ifrån ??, anledningen var främst att det inte krävdes eftersom pjäsernas olika rörelser är så pass för definierade.



Figur 4.9: Flödesschema över den raka rörelsen.

Hur den raka rörelsen fungerar är genom att initiera startpositionen och loopa tills startpositionen har blivit slutpositionen. Inom loopen stegas den nuvarande positionen i rätt riktning och kollar om nuvarande positionen har några pjäser brevid sig, alltså om fler elektromagneter än bara det nuvarande stegets elektromagnet behöver sättas på.

Hur man sätter på en elektromagnet handlar om att skicka ut rätt värde på rätt shiftregister. I Arduino-kortet finns det en variabel på 8 bitar för varje shiftregister (36 stycken), vilket innebär att varje elektromagnet motsvaras av en bit i en variabel, hur dessa värden läggs in är genom att antingen addera $2^{\text{positionen i shiftregistret}}$ till shiftregistrets variabel för att aktivera eller att subtrahera samma värde för att deaktivera.

Vidare så när både nuvarande, och intilliggande (med pjäser på) positioners värden läggs till i respektive shiftregister variabel klockas dessa ut för att aktivera elektromagneterna och flyttar därmed pjäsen ett steg. Därefter ligger en fördröjning för att se till att pjäsen når fram innan de tidigare värdena subtraheras och klockas ut för att deaktivera samma elektromagneter som tidigare aktiverades. Sedan återgår programmet till toppen av loopen och kollar om det steg som just genomfördes resulterade i slutpositionen för att annars fortsätta rörelsen.

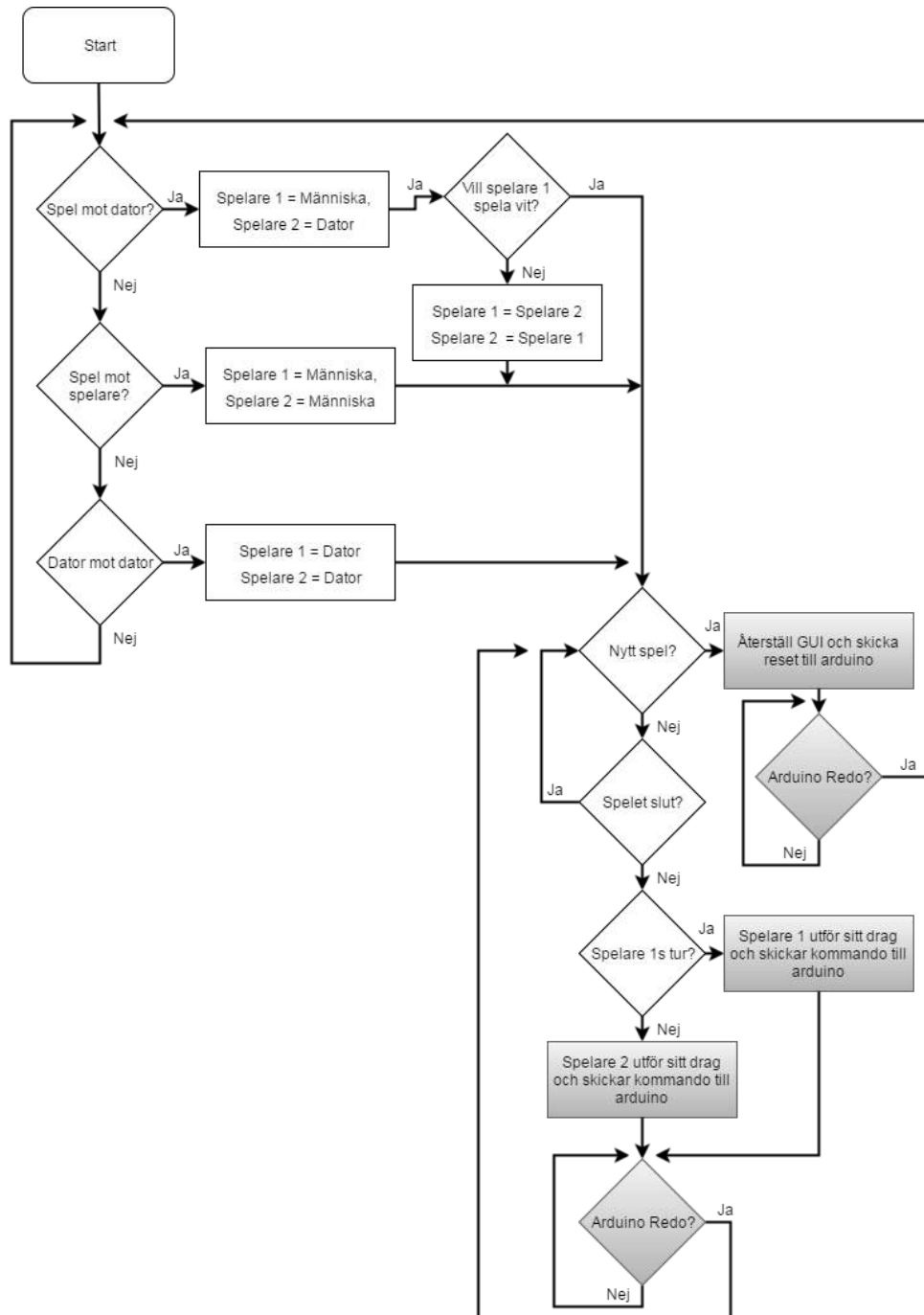
4.2.2 Java software

Själva mjukvaran för schacket och dess logik var ingenting som skulle fokuseras på i projektet som tidigare nämnt i avsnittet om förutsättningar 1.3. Därmed baserades projektets mjukvara på en open source schackmotor skrivet i Java kallat *Carballo* som även har ett tillhörande användargränssnitt som utvecklats med fokus på enkelhet och tydlighet som kan ses i figur 4.10, vilket gjorde det till ett perfekt val för att modifieras inom projektet. Dessutom har schack motorn en maximal svårighetsgrad som jämför sig med toppspelare i världen med en så kallad ELO-rating på 2300, vilket går väl inom det krav på schackmotorn från kravspecifikationen 2.14.



Figur 4.10: Skärmdump från det modifierade användargränssnittet för schackmotorn *Carballo*.

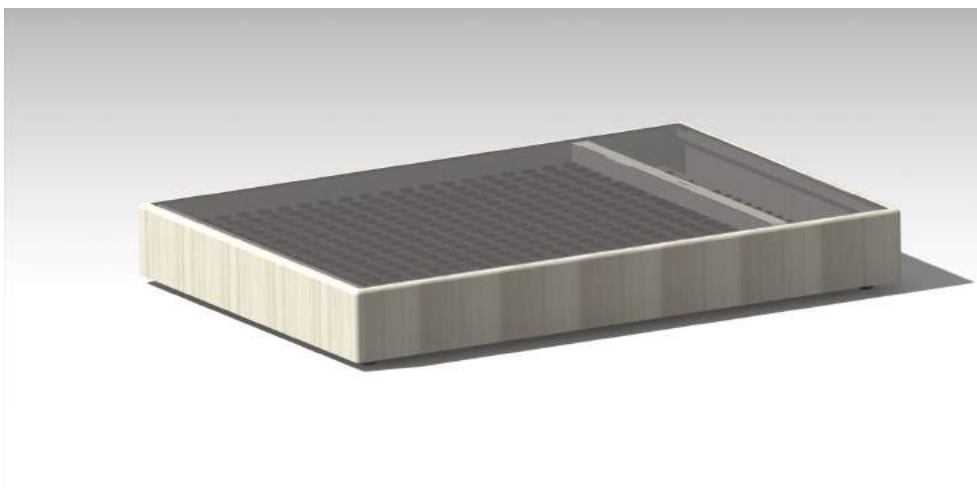
På nästa sida följer ett flödeschema över Carballos användargränssnitt, figur 4.11 där de gråa rutorna är de tillägg som gjorts för att sköta kommunikationen till och från Arduino-kortet. Först väljs vilka som skall spela partiet med hjälp av alternativ i användargränssnittet, och om en spelare väljs som dator går dess drag via Carballos schackmotor. Vidare så görs en kontinuerlig check på om det skall startas ett nytt spel och skickar i så fall ett reset kommando på den seriella utgången (USB) som Arduino-kortet därmed tar emot, därefter väntar java programmet tills Arduino-kortet skriver att den är redo för ny input. Annars loopar programmet igenom och låter varannan spelare utföra sitt drag som skickas ut på den seriella utgången tills dess spelet är över.



Figur 4.11: *Simplifierat flödesschema över Carballo med utökning som kommunicerar med Arduino-kortet.*

4.3 Schackbrädets konstruktion

Schackbrädet har en enkel konstruktion som består av tre lager. Första lagret, som är längst ned, består av en bottenplatta som håller ihop de andra lagren. Andra lagret består av experimentkort varpå alla elektriska komponenter är monterade. Det tredje lagret är den översta ytan av schackbrädet, det vill säga spelplanen. I figur 4.12 nedan syns en rendering av CAD-modellen, där den har modellerats med väggar och en ribba som håller upp det översta lagret. I figur 4.13 syns en rendering av de tre lagren som utgör schackbrädet, modellerat utan skruvar och kablage. Modellerna har gjorts i CAD-programmet Catia V5.



Figur 4.12: *Schackbrädet modellerat med träsidor.*

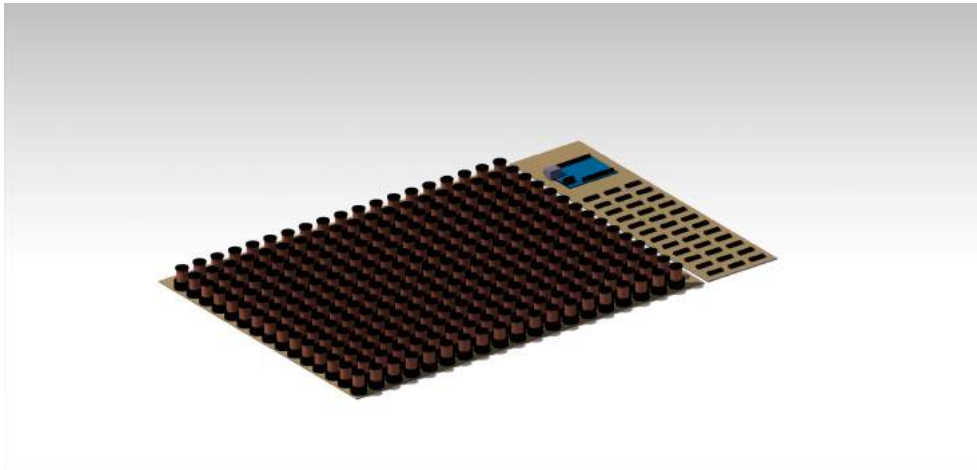


Figur 4.13: *CAD-modell av de tre lagren utan sammanhållande skruvar.*

Första lagret, som är grunden för hela brädet, består av en 460x660x16 mm stor MDF-skiva (Medium Density Fiberboard) som är ett trämaterial tillverkat av små ihoppressade och limmade träfibrer. På undersidan av bottenplattan finns fyra gummifötter som lyfter brädet från underlaget den står på. I plattan finns 40 stycken $\varnothing 4$ mm hål där $\varnothing 4 \times 40$ mm skruv är fästa med tillhörande muttrar. Det är på dessa skruvar som lager två är monterat, 20 mm ovanför bottenplattans övre sida, med hjälp av en kvadratisk mutter

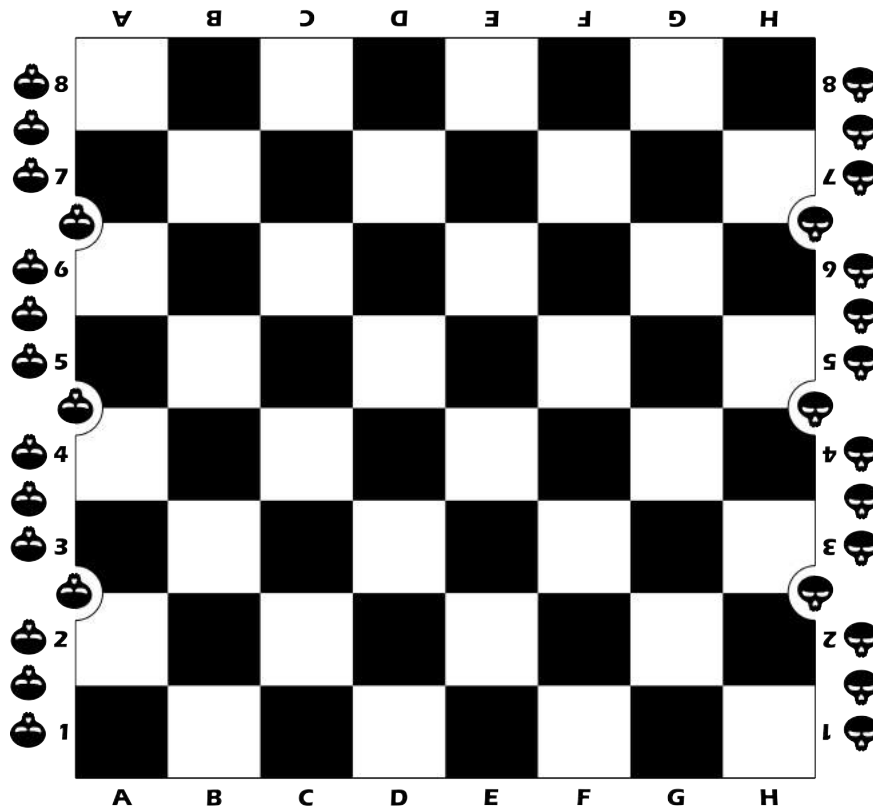
med plastdistans på undersidan av experimentkortet, där det finns kopparbanor som behöver isoleras från muttern, samt en låsmutter på ovansidan av experimentkortet som håller dem på plats. I utkanterna av bottenplattan finns sex st $\varnothing 5$ mm hål. I dessa sitter $\varnothing 5 \times 100$ mm gängstavar monterade med fyra tillhörande muttrar till varje gängstav. Två av muttrarna används för att fästa gängstaven i bottenplattan, och de andra två för att fästa schackets tredje lager, på en höjd av ca 60 mm ovanför bottenplattan.

Andra lagret består av fem stycken avlånga (100x500x1.5 mm) experimentkort. Dessa består av epoxi-papper med raka banor av koppar och små hål i längsriktningen av plattan. Fyra av dessa experimentkortet är placerade med långsidorna mot varandra så att de tillsammans bildar ett 400x500 mm stort bräde. På detta brädet är 285 elektromagneter fästa med en tums (25,4 mm) avstånd från centrum till centrum så att de bildar en matris av elektromagneter med 19 kolonner och 15 rader. I anslutning till varje elektromagnet finns en diod, en Mosfet-transistor och en kabel monterat kring den som finns tydligare förklarad i avsnitt 4.1. Den sista experimentkortet är avklippt så den har dimensionen 100x400 och ligger med långsidan placerad emot de andra experimentkortens ena kortsidor. På detta experimentkortet finns 36 shiftregister monterade med tillhörande kablage. Dessa är kopplade med en kabel till elektromagneterna som är fästa på de andra fyra experimentkortet. På detta experimentkortet finns också mikrokontrollern Arduino UNO monterad, som är kopplad till shiftregistren. I figur 4.14 nedan syns en renderad bild av CAD-modellen för experimentkortet med magneter, shiftregister och mikrocontroller-kortet Arduino UNO monterat. Hålen och kopparbanorna samt resterande elektronikkomponenter och kablar syns alltså inte i bilden.



Figur 4.14: CAD-modell av de fem experimentkortet med magneter, shiftregister och Arduino UNO monterat.

Tredje lagret består av en 460x660x4 mm skiva transparent polystyren vilket gör att det går att se igenom brädet för att se tekniken nedanför. På undersidan av denna har spelplanen, det vill säga schackrutorna och platser för utslagna pjäser, bränts in med en laserskärare, så att svarta rutor ser frostade ut och vita rutor är genomskinliga. I figur 4.15 nedan syns en bild på spelplanen, där döds kallarna ligger på de platser som utslagna pjäser kommer att placeras på. Anledningen till att tre av platserna på varje sida ligger på kantlinjen istället för utanför är för att slippa aktivera lika många elektromagneter som håller fast andra utslagna pjäser när en pjäs flyttas till eller från dödhanteringen.



Figur 4.15: Bild på hur graveringen ser ut, det som är svart i bilden är frostat på det färdiga brädet.

4.4 Konstruktion av pjäser

För att en pjäs ska passa till schacket finns flera begränsningar. För det första får inte pjäsen vara bredare än att den kan åka mellan två andra pjäser utan att stöta emot dem. Detta innebär att maxbredden för en pjäs är en tum (25.4 mm), men för att ha lite marginaler bör pjäsen inte vara större än 20 mm i diameter. En annan begränsning är att det måste finnas en permanentmagnet eller magnetiserbart material i pjäsen så att den reagerar på elektromagneternas magnetfält. Detta har förklarats under avsnitt 3.5.2 och då bäst rörelse har lyckats med permanentmagneter har det valts att använda små neodymmagneter i botten av varje pjäs. Två olika storlekar har använts, till större pjäser har en neodymmagnet med dimensionerna $\varnothing 8 \times 3$ mm använts och till mindre pjäser har tre neodymmagneter med dimensionerna $\varnothing 5 \times 1$ mm använts. En tredje begränsning är pjäsens vikt. Pjäserna måste vara väldigt lätta så att elektromagneterna orkar dra pjäserna över brädet men inte så lätta att de inte klarar av att stå på två intilliggande elektromagneter utan att repelleras iväg från varandra.

Som pjäser har flera alternativ undersökts, bland annat träpjäser och plastpjäser som används till vanliga schackbräden. Vid dessa tester upptäcktes ett väldigt intressant fenomen. Pjäserna i plast hade ett hål i botten i vilka magneterna placerades utan att fästas med hjälp av lim eller tejp. Detta tillät magneterna att röra sig och därmed vrida sig efter magnetfältet vilket gjorde att de var mycket lättare att flytta än pjäser med magneter som satt fastlåsta. Tyvärr var både plast och träpjäserna något för tunga för att förflyttas med hög pålitlighet och hade alldeles för bred bas för att tillåta förflyttning mellan rutor.

Ett alternativ som fungerade var att ta små pjäser från ett magnetiskt rese-schack och byta ut deras ferromagneter mot tre av de små neodymmagneterna. En bild på ett rese-schack med små magnetiska pjäser syns i figur 4.16 [25]. Dessa pjäserna var tillräckligt små och lätta att de följde med magneternas rörelse på brädet. Eftersom de mindre neodymmagneterna kunde användas var inte repelleringen ett problem, men pjäserna är lite mindre i förhållande till schackets rutor än vad som vore önskvärt.



Figur 4.16: Bild på ett magnetiskt schackbräde och små pjäser med magneter i sig.

Det alternativ som projektgruppen satsade på var att 3D-skriva pjäser av plast. På så sätt skulle man kunna välja precis hur stora pjäserna ska vara samt hur de ska vara formgivna. Flera olika försök att använda 3D-skrivna platspjäser har gjorts, där speciellt en lättviktspjäs har använts. Denna är ursprungligen designad och upplagd av Joey Muirhead som öppen källkod på hemsidan Thingiverse.com [26], där folk gratis delar olika 3D-modeller. Dessa pjäser är sedan konverterade från filformatet .stl till formatet .cat-part för att kunna modifieras till att innehålla en $\varnothing 8 \times 3$ mm stor neodymmagnet i botten och sedan utskrivna med en Makerbot Replicator 2X. Till en början konstruerades dessa så att magneten var fastsatt i ett litet hålrum i en ganska tunn botten utan att kunna röra sig. Kungen, som är den största pjäsen, vägde efter flera försök som minst 1.6 g, vilket var något för tungt för att pjäsen skulle flyttas varje gång på ett underlag av ren polystyren. Eftersom det fungerade dåligt gjordes istället basen av pjäsen större och även hålrummet större så att magneten i basen skulle kunna rotera till viss del. Detta gjorde att pjäserna drogs bättre till elektromagneterna, men om hålrummet var så stort att magneten kunde vända på sig attraherades plötsligt två intilliggande pjäser istället för att repellera varandra. Därför gjordes hålet så stort att magneten kunde vrida sig ca 60 grader åt varje håll och alltså inte kunde vända sig.

Pågrund av att pjäserna hade större bas var de även tyngre och dessutom gled de dåligt på underlaget av polystyren som skulle användas till spelplanen. För att minska friktionen och på så sätt förbättra pjäsernas glidförmåga gnedes spelplanen och pjäsernas undersida in med potatismjöl, vilket hade stor inverkan. Därefter gled pjäserna bra och alla rörelser fungerade säkert. Däremot noterades det att de små pjäserna som hade mindre bas, speciellt bonden, kunde välta för att rörelsen var så ryckig och snabb. Detta åtgärdades genom att använda kungens storlek på bas vilken var den största till alla pjäser.

De slutgiltiga pjäserna gjordes i två delar: en $\varnothing 20 \times 6$ mm stor bas som innehåller ett $\varnothing 10 \times 5$ mm stort hål där magneten ligger samt ett grunt hål med större diameter ovanför, och en topp-del som gestaltar vilken pjäs det är. CAD-modellerade bilder på basen och toppen syns i figur 4.17a respektive figur 4.17b. Det större hålet används för att hålla topp-delen på rätt plats. Anledningen att de gjordes som två delar var dels att det går att byta eller ta ut magneten men främst tillverkningstekniskt eftersom det då gick att få en helt plan yta på toppdelens undersida vilken fungerade som lock för hålet till magneten i basen, då den typen av 3D-skrivare som användes inte kan skriva material platt rakt ut i luften. Basen och topp-delen sattes sedan ihop genom att trycka ihop dessa så toppen klämdes fast i basen och en bild på detta syns i figur 4.17c.

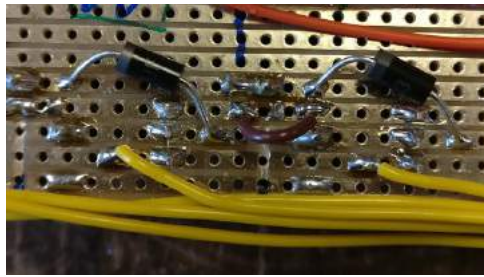


(a) Bild på topp-delen av (b) Bild på basen till kungen (c) Bild på en kung där bas och kungen hos de slutgiltiga pjäserna. topp är ihopsatta.
serna.

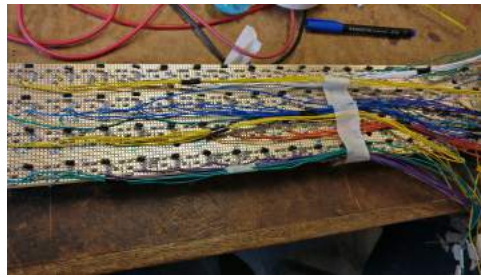
Figur 4.17: CAD-modellerade bilder på topp-delen och basen till de slutgiltiga pjäserna samt en verklig bild på en ihopsatt pjäs.

4.5 Montering av elektronik

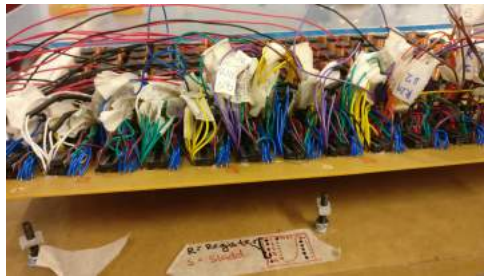
De elektriska komponenter som schacket består av beskrevs i avsnitt 4.1.1 och i avsnitt 4.1 presenterades hur kretsen skulle se ut. Detta realiserades och monterades på fem st experimentkort. Experimentkorten har kopparbanor i längsgående riktning, vilka skärs av där avbrott behövdes, se figur 4.18a. Vissa kopparbanor hålls intakta för att vara gemensam jord (0 V) eller gemensam +5 V vilket innebär att alla elektromagneter på samma rad är parallellkopplade. Alla komponenter har lötts på för hand. Ungefär 3750 lödpunkter har lötts och 1500 avbrott har skurits, totalt är det knappt 900 komponenter som har kopplats ihop. Hur kopplingen blev kan ses i figurerna 4.18b, 4.18c och 4.18d.



(a) Skurna avbrott i kopplingsplattan.



(b) Koppling undersida.



(c) Kopplingsbräda för shiftregister.



(d) Rutnät av elektromagneter.

Figur 4.18: *Montering av komponenter.*

4.6 Verifiering och felsökning av hårdvaran

När det lötts ihop 900 komponenter och flera tusen lödpunkter finns det mycket som kan gå fel, vilket vissa saker också gjorde. För att arbetet skulle ske effektivt löddes alla komponenter av samma sort på en gång, dvs först löddes alla drosslarna, sedan alla transistorer och så vidare. Det gjorde att när hela uppkopplingen på de fyra korten med drosslar inte fungerade som det var tänkt fanns många möjliga felkällor. Att det fanns fel i kopplingen upptäcktes då resistansen mellan den gemensamma jorden och spänningskällan mättes på respektive rad. När transistorerna inte är aktiverade ska resistansen vara såpass hög att ingen ström kan passera, det ska alltså vara som en öppen krets. För drygt hälften av raderna upptäcktes det att resistansen var alldeles för låg och eftersom varje rad innehåller 19 uppsättningar av drosslar, transistorer, dioder och sladdar var det inte helt trivialt att felsöka.

Det första testet var, som redan nämnts, ett öppet kretsprov. Resistansen på den öppna kretsen mättes. Testet ansågs lyckat ifall Ohmmetern visade OL (Open Loop). Ohmmetern kunde mäta upp till 110 M Ohm innan det visade OL och om raden klarade testet kopplades raden in till 5 V med ett nätaggregat, samma spänning som vid användning av schacket. Med en Amperemeter mättes strömmen och den förväntades vara 0 A mellan drain och source då ingen transistor leder. Om ingen ström flödade i kretsen så testades aktivering av varje drossel en i taget, genom att lägga 5 V på gaten. Då en drossel ledde skulle ca 1 A dras från nätaggregatet och drosseln magnetiseras. Om allt detta stämde godkändes raden.

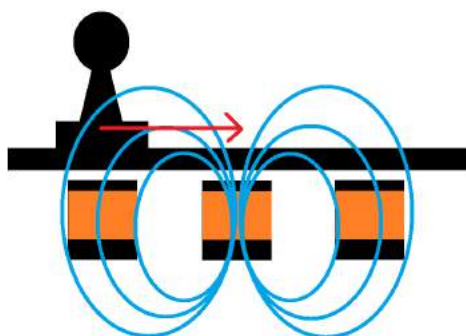
För alla de rader som inte godkändes så började de inspekteras noga. Denna inspektion inleddes genom observation. På vissa ställen fanns tydliga kortslutningar, med lödtenn som flytit ut för mycket, eller så var vissa lödpunkter dåliga och de upptäcktes i samma vea. Lödtenn på oönskade platser skrapades bort, eller samlades upp med hjälp av en lödfläta. En multimeter användes för att hitta förväntade eller oväntade OLs, såväl

hög-resistiva som låg-resistiva kortslutningar. På detta sätt hittades mindre uppenbara utbredningar av lödtenn och dessutom ett antal trasiga transistorer. Eftersom det var långa rader med en hel del gemensamma punkter var det trots mätningarna inte alltid uppenbart vad felet var. Det som gjordes då var att raden delades upp genom att kopparbanorna skars isär, kontrollerade på vilken eller vilka delar felet fanns och fortsatte att dela upp tills den felaktiga sektionen kunde identifieras. Den felaktiga sektorn undersöktes och felet kunde åtgärdas och funktionen kontrolleras. Raden bryggades sedan ihop med lödtenn. Det var ett slitsamt arbete som krävde gott om tålamod, envishet och det tog mycket tid ifrån slutprocessen att åtgärda alla dessa fel.

Experimentkortet med shiftregister kontrollerades ihop med mjukvaran och Arduino-kortet innan det kopplades ihop med alla drosslar. Denna kontroll gjordes genom att spänningen på benen mättes. När aktiveringkommandot för shiftregistret skickades, observerades det att benen gick från att ge ut 0V till 5V, vilket var det förväntade resultatet och alltså fungerade kortet som det skulle. Felsökning är något som kan förväntas krävas vid den här typen av projekt och det blev tyvärr väldigt mycket av eftersom så extremt många komponenter var inblandade. För att minska mängden felsökning borde komponenter löts på och testats efterhand. På så sett skulle felet upptäckas och kunna åtgärdas direkt utan någon lång tid spenderad på felsökning.

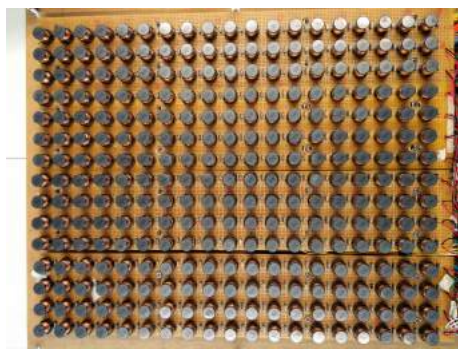
5. Resultat

Målet med projektet var att konstruera och bygga ett fysiskt schackspel där pjäserna kan röra sig automatiskt efter användarens eller datorns kommandon. En prototyp har konstruerats som baseras på ett koncept för förflyttning där ett rutnät av elektromagneter placerats under spelplanen. Pjäser har 3D-skrivits i plast och innehåller en liten permanentmagnet i botten av pjäsen. För att utföra en förflyttning aktiveras elektromagneterna i sekvens vilket innebär att den pjäs som ska flyttas hoppar från elektromagnet till elektromagnet. Detta illustreras i figur 5.1, där de blå linjerna motsvarar ett magnetfält som kommer från den aktiva elektromagneten, vilket gör att pjäsen som står ovanför en intilliggande elektromagnet attraheras och förflyttas till den nya positionen (röd pil).

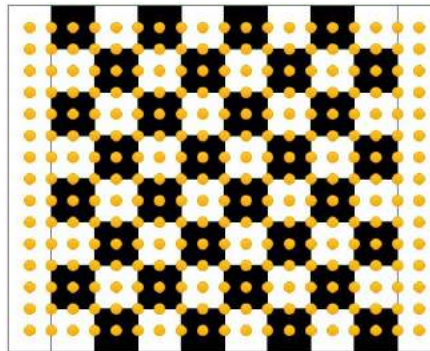


Figur 5.1: Tvärsnitt av konstruktionen som visar hur en pjäs rör sig i ett magnetfält från en avstängd elektromagnet till en aktiverad.

Prototypen består av 285 elektromagneter i ett rutnät som består av 15 rader och 19 kolumner varav två kolumner på vardera sida är allokerade för hantering av utslagna pjäser, och resten av kolumnerna i mitten är till för själva 8x8 spelplanen. En bild på rutnätet av elektromagneter visas i figur 5.2a och i figur 5.2b visas en tecknad bild som förtydligar spelplanen och dödahandlingens position över elektromagneterna, vilka symboliseras av de gula prickarna i figuren. Alla magneter styrs av ett Arduino-kort och kan aktiveras var för sig. Inmatning för att utföra förflyttningar sker via ett användargränssnitt till schackmotorn *Carballo* på en dator som kopplas till Arduino-kortet. *Carballo* är en open source schackmotor, och det är den som är implementerad för att erbjuda användaren möjligheten att spela mot en dator som motspelare.



(a) Bild ovanifrån som visar rutnätet med elektromagneter.



(b) Tecknad bild ovanifrån som tydligt visar spelplanens och dödhanteringens placering i rutnätet med elektromagneter.

Figur 5.2: Två bilder som visar rutnätet av elektromagneter.

Pjäserna som används är 3D-skrivna och består av två ihopsatta delar, en bas med en $\varnothing 8 \times 3$ mm neodymmagnet i sig samt en topp som visar vilken sorts pjäs den är. På spelplanen och pjäsernas undersida är potatismjöl påstruket för att förbättra pjäsernas förmåga att glida på brädet. I figur 5.3 syns schackbrädet med pjäserna uppställda i startposition.



Figur 5.3: Bild på hela schackbrädet.

5.1 Måluppfyllnad

För utvärdera måluppfyllnad jämfördes prototypen mot de krav som ställts upp i kravspecifikationen enligt de verifikationsmetoder som bestämdes i början av projektet. Resultaten, kraven och verifikationsmetoderna går att se i figur 5.4. Eftersom schackbrädet färdigställdes i ett väldigt sent skede har inte alla krav utvärderats exakt och en del buggar inte åtgärdats. Det hade också behövts mer tid åt att kalibrera brädets höjd över magneterna samt pjäsernas utformning och vikt.

Kravspecifikation				
Krav	Mål	Verifikationsmetod	Resultat	Godkänt (Ja/Nej)
1. Prestanda				
a. Tid för en pjäs förflyttning	<5 s	Tidtagning för det längsta draget	Gör tre magnethopp per sekund. Gör därför den längsta diagonala rörelsen på 9 sekunder	Nej
b. Svårighetsgrad	>1500 ELO	Kontrollera specifikationer för schackalgoritmen	Schackalgoritmen klarar upp till 2100 ELO	Ja
c. Ljudvolym	<60 dB (Under normal samtalsvolym)	Mätning av ljudnivå	52 dB uppmätt med mobiltelefon	Ja
d. Elförsörjning	Klara svenskt eluttag	Testa att koppla in det i ett eluttag	Fungerar	Ja
e. Placering inom rutor	Pjäsen ska vara inom ramarna för varje ruta	Kontrollera att alla pjäser står innanför rutorna under 5 schackpartier	Pjäserna landar alltid inom sin ruta förutom vid vissa tillfällen där de ej dras ett magnetsteg eller då de dras fast till en annan pjäs	Nej
f. Förflyttning	Pjäsförflyttning ska utföras utan att slå i andra pjäser	Kontrolleras under 5 schackpartier	Stöter under normal förflyttning inte i andra pjäser men vid enstaka fall dras två pjäser ihop sig när en av pjäserna passerar den andre	Nej
g. Identifierbarhet	Alla pjäser ska vara enkla att identifiera	Fråga person utanför projektgruppen om den kan identifiera pjäserna	Lätt identifierbara	Ja
2. Storlek				
a. Brädets volym	<1000x1000x1000m	Mät prototyp	460x660x110mm	Ja
b. Pjäsens volym	Alla pjäser ska få plats på brädet	Kontrollera att alla pjäser får plats vid startposition av spel	Pjäserna får plats.	Ja
3. Vikt				
a. Vikt för hela prototypen	<20 kg	Väg prototypen	11.3 kg	Ja
b. Vikt för en pjäs	<100 g	Väg pjäserna	Minsta pjäs (bonde): 1 gram Största pjäs (Kung): 2.4 gram	Ja
4. Pris				
Totalkostnad för projektet	<10 000 kr	Beräkna kostnad av samtliga inköp	Uppskattat till 5000kr, på grund av att projektgruppen inte vet hur mycket Chalmers betalar för sina beställningar	Ja
5. Övrigt				
a. Realisbar inom tidsramarna	Fungerande prototyp innan arbetet är slut.	Testa om det går att spela schack med prototypen	Prototypen fungerar. Går att spela människa-människa, människa-dator eller dator-dator	Ja

Figur 5.4: *Kravspecifikation med resultat för prototypen.*

Brädets ljudvolym ligger under normal samtalsvolym vilket är cirka 60 dB [28]. Eftersom konceptet innefattar ett stort antal elektromagneter påverkas både totalvikt och totalvolym väsentligt ifall olika varianter av elektromagneter eller konfigurationer används. Den volym som brädet har är likvärdig med de liknande lösningarna som undersöktes och vikten är med sina 11.3 kg knappt dubbelt så tung som schackbrädet Square Off [8] vilket anses godkänt då det kan bäras av en person.

Tiden för en förflyttning är kravsatt till att alltid vara mindre än 5 sekunder. Detta uppfylls inte, främst eftersom förflyttning av flera pjäser samtidigt inte är implementerat. Då en pjäs slås ut behöver därför den utslagna pjäsen först flyttas till dödhanteringen vid sidan av brädet innan draget kan utföras. I nuläget rör sig pjäserna 3 magneter per sekund. Detta innebär att de flesta dragen sker på under 5 sekunder men att det längsta draget, diagonal förflyttning från ett hörn till motstående hörn tar nio sekunder. Tiden för hur länge varje magnet är aktiverad kan lätt ändras i Arduinokoden men snabbare förflyttning kräver bättre kalibrering av pjäser och spelplanens höjd över magneterna för att säkerställa att pjäserna alltid hinner flytta på sig. Med befintliga inställningar fungerar de allra flesta dragen men ibland fastnar pjäsen och kravet på att pjäserna alltid ska placeras sig inom målrutorna kan därför inte anses helt uppfyllt. Fördröjningen i Arduinokoden har experimenterats med och det har gått att snabba på rörelserna avsevärt, tyvärr så blir pålitligheten hos dragen mindre och pjäserna börjar bete sig olikt varann.

Att pjäserna beter sig olika är på grund utav att deras vikt varierar sinsemellan, om exempelvis fördröjningen dras ner till 100ms kommer bönderna kunna flyttas bra medan kungen inte kan röra sig alls.

Ett krav på förflyttningen är att pjäserna inte skall slå i andra pjäser under spelets gång. Detta är återigen något som uppfylls vid de allra flesta drag men som vid enstaka tillfällen misslyckas då pjäser fastnar på varandra. Majoriteten av dessa fall kan troligen åtgärdas genom relativt lätta åtgärder. Det främsta är att inte låta den första utslagna pjäsen gå mellan raderna av pjäser vid spelets start. Det andra är att det är nästan enbart bönder som fastnar på andra pjäser, dessa skulle därför behöva göras något större och tyngre för att förhindra detta.

Pjäserna som används har tydligt utseende och det går lätt att identifiera olika typer av pjäser. Kravet på att pjäserna ska ha en vikt på under 100 gram är också uppfyllt, då de endast väger ett par gram för att vara möjliga att flytta med hjälp av elektromagneterna.

Den högsta möjliga svårhetsgraden hos schackmotorn är kravsatt till minst 1500 ELO. Den schackmotor som används, *Carballo*, har en högsta ELO på 2300 vilket är högt över det uppsatta kravet. Det tillhörande användargränssnittet tillåter endast godkända förflyttningar, vilket är en form av fuskindikator som var en önskad extrafunktion.

Kostnaden för alla komponenter använda i slutkonstruktionen är beräknade till cirka 3000 kr. Utöver detta har cirka 2000 kronor lagts på komponenter och material för de tester som genomförs under projektets gång. Exakta kostnader är svåra att beräkna då projektgruppen inte har tillgång till några fakturor och inte vet vilka rabatter och förmåner Chalmers har vid inköp. Kravet på under 10 000 kronor är med säkerhet uppfyllt.

5.2 Slutsats

Syftet med projektet var framför allt att i utbildningssyfte utveckla en teknisk lösning med ett schackbräde som kan flytta schackpjäser efter spelares kommandon. Även om prototypen inte är helt optimerad har schackpartier spelats och pjäser flyttats efter användarens och datorns kommando, vilket anses uppnå det huvudsakliga målet med projektet. Gruppen har under projektets gång ställts inför många nya utmaningar och lärt mycket, framförallt genom diskussion med kunniga personer och praktiskt arbete med elektronik. Det slutgiltiga konceptet har både fördelar och nackdelar men skulle med fortsatt utveckling och förfining kunna utvecklas till en färdig produkt. Denna kan troligen konkurrera med produkter med samma funktion men baserade på andra koncept, såsom exempelvis *Square Off*. Vidare resonemang kring projektet och dess resultat förs i kapitel 6, Diskussion.

6. Diskussion

Diskussion och gruppmedlemmarnas tankar kring projektet har delats upp i två delar. En del som hanterar lärdomar och erfarenheter. Här diskuteras vad gruppen hade gjort annorlunda om samma projekt hade gjorts igen. Den andra delen hanterar rekommendationer för fortsatt utveckling av den prototyp som togs fram i detta projekt.

6.1 Vad hade kunnat göras annorlunda

En sak som tog onödigt lång tid under projektet var att försöka göra egna elektromagneter, samt att dimensionera dessa. Om vi hade pratat med Robert Karlsson vid institutionen för Elektroteknik som tipsade om drosslarna tidigare hade vi kunnat börjat med testningen av dessa mycket tidigare och därmed säkert hunnit längre i projektet.

I ett tidigt skede skulle beräkningsprogrammet Comsol multiphysics ha kunnat användas för att beräkna kraften som elektromagnetens magnetfält påverkar pjäsen med, beroende på till exempel avstånd, material och utformning av elektromagneten. Detta hade möjliggjort att fler olika sorters konfigurationer hade kunnat testas på ett billigare och snabbare sett. En intressant sak att testa hade varit att försöka styra elektromagnetens magnetfält, framförallt genom olika former av överdelar till elektromagneterna eftersom en metallbricka på toppen gav upphov till ett mer spritt magnetfält.

Comsol användes men upptäcktes för sent för att använda det tillräckligt ordentligt för att stå som grund till valet av elektromagneter. Ett annat problem med detta är att det vore svårt att sedan hitta lämplig elektromagnet som motsvarar det som fås ut av Comsol beräkningarna. Det finns mycket elektronikkomponenter, men det är svårt att hitta precis det man vill ha. En utformning som skulle vara intressant att testa, till en början i Comsol, är en så kallad pot-core kärna. Kärnan ser i genomskärning ut som ett liggande E. En pot-core kärna är ett sätt att se till att luftgapet blir mindre och dessutom ser magnetfältet annorlunda ut så det vore intressant att se hur stor skillnad detta skulle göra. Istället för att använda Comsol gjordes tester, dels av egentillverkade elektromagneter, men även av ett tiotal drosslar. Testerna var utförliga och gav oss större förståelse av vad som påverkar en elektromagnets styrka. Det hade varit bra om vi standardiserat testerna mer och framförallt hade haft ett bättre sett att samla resultaten på.

Det som krävde mest tid under själva konstruktionen av brädet var att löda ihop alla komponenter, skära sönder kopparbanor och framförallt felsöka de kopplingar som blivit fel, vilket nämdes i 4.5. Komponenterna och lödningen borde testats efter att varje komponent löts fast för att tidigt upptäcka fel. För att minska antalet fel och för att snabba på processen hade vi möjligen kunnat skriva ut egna kretskort som var anpassade specifikt efter vårt ändamål. Detta hade minskat antalet lödpunkter eftersom det blir färre bryggningar. Dessutom hade det inte varit lika många långa sladdar att hålla reda på och få in på så liten yta som vi behövt göra. Alternativt hade man kunnat göra det mer modulärt i kopplingarna för att kunna testa enskilda delar lättare.

Vi var väldigt inställda på att använda en genomskinlig spelplan för att synliggöra all teknik som ligger bakom förflyttningarna. Vad som dock upptäcktes i efterhand och inte togs i beaktning under projektets gång var hur friktionen var på det bräde som valdes. Vi valde att ha ett bräde i transparent polystyren, istället för denna plast hade vi kunnat kolla på material med lägre friktionskoefficient för att pjäserna lättare skulle kunna flytta på sig. Friktionskoefficienten för pjäserna skulle också kunna påverkas om man väljer ett annat material.

Något annat som hade kunnat undersökas mer är hur stort man vill att brädet ska vara. Vi utgick från vad som verkade vara ett normalstort schackbräde och dimensionerade därför bräde, pjäser och elektromagneter efter det. Det finns även mindre drosslar som är lite billigare som borde kunna sitta mycket närmare varandra och på så sätt skapa en mindre spelplan. De magneter som användes som pjäser i de tester som genomfördes repellerades för starkt för såpass små avstånd, men det är möjligt att svagare magneter kan användas. Vi är ändå nöjda med de val vi gjorde men det hade varit intressant att undersöka hur mycket spelet kan skalas ner och fortfarande fungera.

6.2 Förslag för fortsatt arbete

Trots att projektet färdigställts finns det flertalet aspekter av projektet som skulle vara intressant att fortsätta utveckla. Det har i föregående avsnitt redan behandlats vad som skulle kunna göras annorlunda, men det finns även en hel del som kan utvecklas vidare från den punkt där projektet slutade.

På grund av att koden till Arduino-kortet bygger på att ta emot input i textform på vilken förflyttning som ska ske på formen a1b1 (som specificerar startruta och slutruta), kan denna input i princip ske på vilket sätt som helst så länge den skickas vidare till Arduino-kortet som förflyttar pjäserna. Exempelvis skulle röst- eller ögonstyrning kunna implementeras, som tidigare skulle detta kunna underlätta för folk med funktionsnedsättning som inte har möjlighet att flytta sina pjäser på annat sätt.

Hårdvaran för projektet är dimensionerad för att kunna sätta på upp till 15 elektromagneter på samma gång, samtidigt är parallell rörelse i mjukvaran inte implementerat. Därmed är det ett starkt förslag för fortsatt utveckling av projektet att implementera detta, då det skulle förhöja användarupplevelsen avsevärt. Anledningen bakom att det inte gjordes från början var på grund utav att Arduino inte stödjer skapandet av trådar i programmeringen, utan koden behöver utökas med antingen så kallade “interrupts” eller “timers” för att kunna hoppa emellan olika processer (alltså att simulera trådar) och därmed kunna utföra flera förflyttningar på samma gång.

Det här projektet har fokuserat enbart på att skapa ett schackspel, men problemet som har lösts är i huvudsak förflyttning av pjäser på ett bräde. Detta innebär att samma princip skulle gå att använda till alla möjliga typer av brädspel. Man skulle då till exempel kunna ha olika spelplaner som man lägger på brädet, samt pjäser för de olika spelen, så skulle man kunna få flera automatiserade spel på samma bräde som exempelvis *fia med knuff* eller *backgammon*.

Eftersom anledningen till detta projekt har varit att göra ett lite roligare och mer in-

tressant schackspel jämfört med att spela på en datorskärm så hade det varit kul att undersöka möjligheterna till interaktion mellan pjäserna mer. Eftersom det hittades ett patent där pjäser kunde svinga en yxa eller använda svärd så vet vi att det finns idéer till hur man skulle lösa en sådan sak. Om pjäserna skulle kunna ”slå sönder” varandra skulle det förhöja wow-faktorn av schackbrädet väsentligt. En idé som kom upp under projektets gång var att ha ytterligare någon typ av elektromagnet som höll ihop pjäserna och som kunde avaktiveras när den blev utslagen. Något sådant hade varit kul att se ifall detta bräde utvecklas.

Syftet med detta projekt har inte varit att skapa en kommersialiserbar produkt, utan projektet har utförts framförallt i utbildningssyfte. Dock är det intressant att kolla på hur man skulle kunna göra detta till en produkt för försäljning. Efter en kort undersökning av marknaden för liknande produkter så finns det en nyare produkt som sticker ut och gör samma sak som vår. Detta schackbräde kallas *Square Off* [8] och är tänkt att kosta \$375-\$450 beroende på modell i detaljhandeln. Fokus i vårt projekt har inte varit att minimera tillverkningskostnader utan på att det ska fungera så snabbt som möjligt. En kort överslagsräkning av komponenterna som ingår i det färdigställda schackbrädet ger ca 3000kr vilket är lite mindre än vad *Square Off* säljs för. Då skulle vi givetvis inte gå med någon vinst, men om detta skulle ske i massproduktion skulle priset gå ner rätt så väsentligt och skulle troligen kunna konkurrera rent prismässigt med *Square Off*.

För att en användare av schackbrädet själv skulle kunna flytta på pjäserna som i *Square Off* så hade det behövts någon typ av sensor på varje ruta av brädet. En faktor som vi i gruppen har tänkt på är att i deras lösning så kräver sensorerna att man trycker ner både där man flyttar från samt dit man vill flytta, detta anser vi vara ett störande moment och hade därför valt att använda oss av andra typer av sensorer som endast registrerar huruvida det står en pjäs på rutan eller inte. På så sätt hade brädet förstått varifrån en pjäs blir upplockad, vilken pjäs det är (eftersom det alltid är samma utgångsläge), och när den ställs ner igen så vet brädet vilket drag som har utförts och kan på så sätt komma med ett motdrag. Detta hade ökat känslan av att man spelar ett fysiskt schack eftersom man själv får röra på pjäserna och inte bara flytta dem via en datorskärm.

Då *Square Off* är helt portabelt så har vi tänkt lite på hur det skulle gå att ordna på vårt bräde. Som nämnts i avsnitt 4.1.1 så hade datorn som kopplas till brädet via en USB-kabel kunnat ersättas av en Raspberry Pi som är en mikrodator som skulle kunna få plats på det befintliga brädet. Dessutom skulle allt kunna kopplas till ett batteri och därmed skulle hela brädet vara portabelt och kunna användas var som helst.

För tips utöver detta samt förslag på olika sätt att lösa alla delproblem som vi inte har behandlat på grund av tidsbegränsningen finns en hel del att tillgå i appendix A.1 som handlar om lösningar på alla delproblem som vi kunde komma på kring schackbrädet.

Litteraturförteckning

- [1] Billwall. (2007). *Computers and Chess - A History*. Hämtad 2017-02-04, från <https://www.chess.com/article/view/computers-and-chess---a-history>
- [2] Schröder, E. (2007). *Chess Genius vs Kasparov*. Hämtad 2017-02-04, från <http://www.top-5000.nl/matches/1994.htm>
- [3] Digital Game Technology. (2017). *HISTORY OF THE DGT BOARD*. Hämtad 2017-01-31, från <http://www.digitalgametechnology.com/index.php/aboutdgt/history/115-history-of-the-electronic-chess-board>
- [4] Wizard's Chess. *Harry Potter Wiki*. Hämtad 2017-05-11, från http://harrypotter.wikia.com/wiki/Wizard%27s_Chess
- [5] Name unknown [FunGowRightNow12]. (2012, 2 april). *Wireless Arduino Powered Chess - Internals* [Videofil]. Hämtad från <https://www.youtube.com/watch?v=XCp4M08IWnA>
- [6] Chess Programming WIKI. (2016). *Milton Bradley Phantom*. Hämtad 2017-05-10, från <https://chessprogramming.wikispaces.com/Milton+Bradley+Phantom>
- [7] Ismenio. (2016). *MB Grandmaster*. Hämtad 2017-05-10, från http://www.ismenio.com/chess_mb_grandmaster.html
- [8] KickStarter. (2017). *Square Off - World's Smartest Chess Board*. Hämtad 2017-05-10, från <https://www.kickstarter.com/projects/infivention/square-off-worlds-smartest-chess-board-relaunched/description>
- [9] Lawrence T. Jones, Ashley Howden, Mark S. Knighton, Anson Sims, David L. Kittinger, Richard E. Hollander. (1981). *Robot computer chess game*. US4398720 A. <https://www.google.com/patents/US4398720>
- [10] Lab Automation Robotics. (2012). *Cleaning Robots*. Hämtad 2017-05-11, från <https://labautomationrobotics.wordpress.com/2012/06/28/cleaning-robots/>
- [11] Name unknown [kipkay]. (2016, 8 januari). *How to Make a Tiny Robot!* [Videofil]. Hämtad från <https://www.youtube.com/watch?v=MEnddcqgg64>
- [12] KMel Robotics [TheDmel]. (2012, 31 januari). *A Swarm of Nano Quadrotors* [Videofil]. Hämtad från <https://www.youtube.com/watch?v=YQIMGV5vtd4>
- [13] Name unknown [Ugoita T.]. (2015, 28 februari). *DANCING PAPER* [Videofil]. Hämtad från <https://www.youtube.com/watch?v=05gPzM70J34>
- [14] Maketechnics. (2015). *BOOGIE BOX: the Electromagnetic Dance Floor*. Hämtad 2017-02-07, från <http://www.instructables.com/id/BOOGIE-BOX-The-Electromagnetic-Dancer/?ALLSTEPS>
- [15] The Libertarian Republic. (2013). *Whack-a-mole*. Hämtad 2017-02-16, från <http://thelibertarianrepublic.com/wp-content/uploads/2013/10/whack-a-mole.jpg>
- [16] BRIO. (2013). *Labyrinth*. Hämtad 2017-02-16, från <http://www.brio.se/Products/Games/games/labyrinth>
- [17] Magnetism. Olof Beckman. *Nationalencyklopedin* Hämtad 2017-05-08, från <http://www.ne.se/uppslagsverk/encyklopedi/lång/magnetism>
- [18] University of Surrey. (2010). *The force produced by a magnetic field*. Hämtad 2017-03-10, från <http://info.ee.surrey.ac.uk/Workshop/advice/coils/force.html>

- [19] Einar Ueland. (2017). *A* (Astar / A Star) search algorithm*. Hämtad 2017-03-10, från <https://se.mathworks.com/matlabcentral/fileexchange/56877-a---astar---a-star--search-algorithm-easy-to-use>
- [20] Panasonic. (2014). *Panasonic Power Inductors Data Sheet*. Hämtad 2017-05-12, från http://www.farnell.com/datasheets/2059834.pdf?_ga=2.29613759.1653628107.1494611407-1419561108.1486721041
- [21] OscarLiang.com. (2014). *HOW TO USE MOSFET-BEGINNER'S TUTORIAL*. Hämtad 2017-05-09, från <https://oscarliang.com/how-to-use-mosfet-beginner-tutorial/>
- [22] Philips Semiconductors. (1998). *INTEGRATED CIRCUITS DATA SHEET*. Hämtad 2017-05-12, från <https://www.arduino.cc/en/uploads/Tutorial/595datasheet.pdf>
- [23] Orsini, L. (2014). *Arduino Vs. Raspberry Pi: Which Is The Right DIY Platform For You?*. Hämtad 2017-02-04, från <http://readwrite.com/2014/05/07/arduino-vs-raspberry-pi-projects-diy-platform/>
- [24] Alberto Ruibal [albertoruibal]. (2016). *Carballo Chess Engine* <https://github.com/albertoruibal/carballo>
- [25] igogo. (2017). *VIICSC Min*. Hämtad 2017-05-09, från <https://des.igogo.es/uploads/2015/201510/heditor/201510281729591713.jpg>
- [26] Joey Muirhead [BigBadBison]. (2014). *Spiral Chess Set*. Hämtad 2017-05-09, från <http://www.thingiverse.com/thing:470700>
- [27] Roman Hegglin [roman_hegglin]. (2013). *Chess Set - Voronoi Style*. Hämtad 2017-05-11, från <http://www.thingiverse.com/thing:172960>
- [28] DBDBDB. (2016). *Ljudnivåer*. Hämtad 2017-05-12, från <http://www.dbdbdb.nu/ljudnivaer/>
- [29] NEONODE INC. (2017). *AIRBAR*. Hämtad 2017-02-05, från <http://www.air.bar/>
- [30] Woodford, C. (2017). *Reed switches*. Hämtad 2017-02-05, från <http://www.explainthatstuff.com/howreedswitcheswork.html>

A. Appendix

A.1 Dellösningar

Nedan beskrivs olika dellösningar för våra delproblem som ligger under huvudproblemet. Delproblemen är rubrikerna, med olika indelningar i underrubrikerna.

A.1.1 Igenkänning av pjäser och dess position

Sensorer

- **Touchsensor**
Inspiration från en typ av kamera som man sätter på laptops för att få touch. Bryter ljus på olika avstånd för att avgöra var pjäserna är. [29]
- **Ljussensor**
Förflyttningar registreras med hjälp av ljussensorer i varje ruta som känner av ifall det är något som står på rutan eller inte.
- **Viktsensor**
Förflyttningar registreras med hjälp av en våg under varje ruta som känner av vilken ruta som tappade vikt och vilken som fick mer vikt. Skulle kunna vara en microswitch eller Miniature snap-action switch. Om varje enskild pjäs vikt är relevant, då behövs en typ av våg.
- **Magnetsensor**
Förflyttningar registreras med hjälp av reed switchar under varje ruta på brädet. [30]
- **Metalldetektor**
Förflyttningar registreras med hjälp av metalldetektorer under varje ruta på brädet. Pjäserna utrustas med någon typ av metall för att detektering skall kunna ske.
- **Strömdetektor**
Varje ruta utrustas med 2 ledande piggar som får kontakt genom någon slags ledning genom pjäserna och agerar på så sätt som en switch för varje ruta och då kan man lätt veta om det står en pjäs på rutan eller inte. Pjäserna skulle även kunna utrustas med olika motstånd för att kunna identifiera vilken pjäs det är.
- **Trycksensor**
Registrerar position när man trycker på rutan. Tryck först på rutan man vill flytta från sedan dit man flyttat.

Lokala system

- **Kamera**
 - Pjäserna har qr-koder på sig och brädet övervakas av en kamera som kan läsa qr-koderna för att identifiera pjäsernas position.
 - Pjäserna har olika färg så att de kan identifieras med en kamera.
 - Kameran känner endast igen vilka rutor som är tomma. På så sätt kan man identifiera vilket drag som gjorts.
- **Antenn.**
Central antenn som känner av elektromagnetiska signaler från sändare i varje pjäs.

Datorminne

- **Minne för pjäserna**

Datorn flyttar alla pjäser, i så fall kommer datorn ihåg var pjäserna är istället för sensorer.

A.1.2 Inmatning för förflyttning

De två första underrubrikerna är för användarinmatning och den sista för AI'ns inmatning

Digital inmatning

- **Röststyrning**

Inmatning med röst till en mikrofon, som sedan får ljudsignalen behandlad i en dator (extern eller dedikerad) och skickar ut kommando till styrsystemet.

- **Dedikerad pekanordning**

Inmatning med hjälp av en pekanordning som är permanent kopplad till brädet, till exempel ett knapp-bräde, en digital skärm, rutnät etc.

- **Appstyrning**

Styrning via app på telefon.

- **Extern dator**

Styrning via vanlig dator med datorprogram.

Manuell förflyttning

- **Handförflyttning**

Att människan själv kan flytta sina egna pjäser.

Schackalgoritm

- **Stockfish**

En open source schackalgoritm som bestämmer drag och skickar signaler till styrsystemet. Denna skulle kunna implementeras på en raspberry pi eller en vanlig dator.

A.1.3 Konstruktion av fysiska brädet

Praktiska lösningar

- **Död-depå**

En plats där utslagna pjäser står/förvaras

- **Förvaringsutrymme för pjäser**

Utrymme där pjäserna förvaras när man inte spelar. Skulle kunna vara automatiserat för att automatiskt förbereda nytt spel (placera pjäserna).

Utseendemässiga löningar

- **Digital skärm**

Brädet är en skärm, vilket gör att man kan anpassa den till vilken schackbakgrund man vill ha, eller alternativt använda en annan spelplan.

- **Projicerat schackbräde**

Schackbrädets rutor är projicerade av en projektor, existerande exempel är projicerade tangentbord.

- **Polariserat glas**

Schackbrädet är konstruerat av polariserat glas. Detta är delvis genomskinligt så att man kan se tekniken under. Polariserat så att rutor syns vid belysning.

- **Frostat glas** Genomskinligt bräde gjort av glas, med rutor gjorda genom frostat/-borstat glas.
- **Genomskinligt bräde**
Brädet är gjort av ett genomskinligt material, till exempel plast, plexiglas eller glas, så att man kan se elektroniken under.
- **Plastbräde**
Brädet är tillverkat av plats.
- **Träbräde**
Brädet är tillverkat av trä.
- **Metallbräde**
Brädet är tillverkat av metall.
- **Utbytbar plan**
På översta lagret av brädet läggs en filt eller skiva. Dessa kan man ha flera varianter av med olika mönster eller teman, så genom att byta ut dem kan man ändra hur spelplanen ser ut.

A.1.4 Konstruktion av fysiska pjäserna

Praktiska lösningar

- **Filt**
Pjäserna har filt under sig för att glida bra och ljudlöst.

Utseendemässiga lösningar

- **Flames**
Pjäserna består av flammor, olika färger och/eller storlekar indikerar olika gubbar.
- **Träfigurer**
Pjäserna är gjorda av trä.
- **Plastfigurer**
Pjäserna är gjorda av plast.
- **Metallfigurer**
Pjäserna är gjorda av metall.
- **Glasfigurer**
Pjäserna är gjorda av glas.

A.1.5 Extrafunktioner

Relaterade till pjäserna

- **Formbara pjäser**
Pjäserna är byggda av ett material så att man kan forma sina egna pjäser, så att man kan förstöra pjäser och bygga upp dem igen. Ett exempel på material är Play Doh-lera.
- **Delade pjäser**
Pjäser är delade från början och hålls ihop med till exempel elektromagneter. När en pjäs blir utslagen går de isär för att visulsera att de dör.
- **Vinnardans**
När ett lag vinner dansar pjäserna.
- **Smältande pjäser**
Pjäserna smälter när de slås ut för att markera att pjäsen dör.

- **Städrobot**
En liten robot åker runt och städar/för bort pjäser när de dör.
- **Fusksensor**
En fusksensor med indikator är implementerad för att känna av ogiltiga drag.
- **Svävande pjäser**
Pjäserna svävar med hjälp av doughnut-formade magneter under brädet. Magneterna under brädet kan förflyttas medan pjäsen följer med. Om magneterna under brädet är elektromagneter kan de stängas av då pjäsen slås ut och då falla ner på brädet.
- **Ferrousfluid**
Pjäserna är gjorda av metall, skruvformade. Planen är täckt av ferrousfluid som klättrar upp på pjäserna då de skall flyttas med hjälp av en elektromagnet. Allt för att schacket skall bli mer effektivt. I övrigt liknande lösning som en "vanlig" med elektromagneter.

Relaterade till brädet

- **Smartfilm**
Ett glasbräde med smart-film. Tanken är då att man enkelt kan välja om man vill ha ett genomskinligt eller ett opakt bräde.
- **Vippa**
Vippa som skjuter iväg pjäserna när de slås ut.

A.2 Undersökning av andra projekt

Sammanställning av artiklar och videomaterial		
Källförteckning	Innehåll	Resultat
Yang, Daylen. 2017. Stockfish - Strong open source chess engine. Stockfish Chess. https://stockfishchess.org (Hämtad 2017-01-24).	En schackalgoritm som är open source ("öppen källkod"), vilket innebär att den är gratis att ladda ner och använda (så länge man refererar till hela källkoden, alternativt refererar till varifrån hela källkoden kan hittas.) . Algoritmen kan användas på både datorer och andra plattformar så som mobiltelefoner och tablets.	I och med att Stockfish innefattas av begreppet öppen källkod, kan den användas till projektet på så sätt att vi ej behöver skriva schackalgoritmen själva.
Bourque, Brad. 2015. Arduino vs. Raspberry Pi: Mortal enemies, or best friends?. Digital Trends. 8 Mars. http://www.digitaltrends.com/computing/arduino-vs-raspberry-pi/ (Hämtad 2017-01-24).	Går igenom skillnader, samt för- och nackdelar mellan minidatorn Raspberry Pi och mikrocontrollern Arduino.	Tips på hur man ska tänka på vid val av minidator/mikrokontroller vid projektets konstrueringsfas.
Hazmat. 2012. Building a Sish Board, a tutorial. Sish - Computer Interfacing DIY Chessboards.[Blogg]. http://sishchess.blogspot.se/p/building-sish-board-tutorial.html (Hämtad 2017-01-24).	Beskrivning av hur man bygger ett eget schack och implementerar schackalgoritmen stockfish med hjälp av en liten dator, i detta fall en raspberry Pi.	Tips på hur man kan implementera en schackalgoritm.
Maxim. 2015.Arduino & Raspberry Pi Chess Computer - Wooden Chess Board with Reed Switch Piece Recognition and LED move indicators. http://chess.forththerapy.co.uk/home/ (Hämtad 2017-01-24).	Beskrivning av hur man bygger ett eget schack som känner igen pjäspositionen samt implementerar schackalgoritmen stockfish med hjälp av Raspberry Pi och Arduino.	Tips på hur man kan implementera en schackalgoritm samt hur man kan göra för att få pjäsposition.
Reed Switch. [u.å.]. Wikipedia. https://en.wikipedia.org/wiki/Reed_switch (Hämtad 2017-01-26).	Reed Switch sensorer som reagerar på magnetfält, samt LED lampor som indikerar strömföring.	Tips på hur igenkänning av pjäser på brädet kan lösas.
Smart Chess board Arduino project (2016) [Video]. Tillgänglig: https://www.youtube.com/watch?v=DB4sVPR6g7o (2017-01-26).	Visar hur applicering av Reed Switch sensorer på ett schackbräde, för att indikera var pjäserna förflyttas.	Tips på hur Reed Switch sensorer kan användas.
Wireless Arduino Powered Chess (2012) [Video]. Tillgänglig: https://www.youtube.com/watch?v=dX37LFv8jWY (2017-01-26).	Visar hur applicering av Reed Switch sensorer på ett schackbräde, för att indikera var pjäserna förflyttas. Använder trådlös överföring via internet för två schackbräden, samt hanterar döda pjäser.	Tips på hur hantering av döda pjäser ska tas hand om.
Wireless Arduino Powered Chess - Internals (2012) [Video]. Tillgänglig: https://www.youtube.com/watch?v=XCp4M08IWnA (2017-01-26).	Visar hur insidan av schackbrädet ser ut, när applicering av Reed switch sensorer har använts på ett schackbräde, för att indikera var pjäserna förflyttas.	Tips på hur Reed Switch sensorer kan användas och hur dessa appliceras på schackbrädet.

Haley Amason, Joshua Burbridge, Brittany Nottingham, Thong Tran. 2012. Magic Chess. Projekt rapport, University of Central Florida, Orlando. http://eecs.ucf.edu/seniordesign/fa2012sp2013/g03/documents/SDCP2.pdf (Hämtad 2017-01-31).	Projekt rapport för en lösning på ett självspelande schackspel.	Tips på hur man kan lösa uppgiften med ett självspelande schackspel, ger kunskap om hur vissa delar kan genomföras samt tips om andra som kan förbättras.
Excalibur Phantom Force Chess - Chess Computer Robot Gameplay (2016) [Video]. Tillgänglig: https://www.youtube.com/watch?v=MNV84Z4j8z4 (2017-01-31).	Ett självspelande schackspel, som förflyttar datorns pjäser.	Ger tips om hur hästens förflyttning kan genomföras (24 sekunder in i videon).
Mephisto Phantom Robot chess computer (2012) [Video]. Tillgänglig: https://www.youtube.com/watch?v=BobOCMj1Qhs (2017-01-31).	Ett självspelande schackspel, som förflyttar datorns pjäser. Använders sig dessutom av en resetfunktion.	Ger tips om hur resetfunktionen kan implementeras.
Lawrence T. Jones, Ashley Howden, Mark S. Knighton, Anson Sims, David L. Kittinger, Richard E. Hollander. (1981) Robot computer chess game. US 4398720 A	Patent på en robotarm som flyttar på schackpjäserna.	Ger tips på hur förflyttning av schackpjäser kan ske.
Licinio Basevi. (1983) Instructional chess game. US 4515371 A	Schackpjäser med speciella egenskaper; de kan svinga yxa, använda svärd etc.	Ger tips på hur man kan förgöra motståndarens pjäser.
Watters, Mike. 2017. Patents. Chess computer UK. http://www.chesscomputeruk.com/html/patents.html (Hämtad 2017-01-31).	Webbsida som samlar många patent för schackbräden som i någon form använder sig av en schackdator.	Ger tips på vad som upfunnits samt hur det utförts.

Figur A.1: Sammanställning av andra projekt och patent

A.3 Pughmatriser

Chalmers		Pughmatris 1 (Relativ beslutsmatris):							
Utfördare:									
Frederic Furborg	Herman Quint	Oskar Hulthén							
Örjan Landgren	Helena Petterson								
Kriterier		Alternativ							
	Vikt	XY-Magnet	r-θ -Magnet	Klomaskin	Robotarm	Små radiostyrda pjäser	Drönare	Elektromagneter i rutnät	Vipp-labyrint
Förflyttningshastighet	5		0	-1	1	0	1	1	-1
Precision	2		0	0	0	-1	-1	-1	-1
Enkelt att programmera	3		-1	0	-1	-1	-1	-1	-1
Enkel tillverkning	2		-1	-1	-1	-1	1	-1	-1
Vikt	2		0	-1	-1	-1	1	-1	-1
Storlek	2		0	-1	-1	1	-1	0	-1
Multi-movement	3		1	0	1	1	1	1	0
Användarupplevelse	5		0	-1	-1	1	1	1	1
Ljudnivå	2		0	0	0	0	-1	1	-1
Kostnad	1		0	0	0	-1	-1	-1	-1
Nettovärde	27	0	-1	-5	-3	-2	0	-1	-7
Viktat nettovärde		0	-2	-16	-6	0	7	5	-14
Rangordning		3				3	1	2	

Figur A.2: Pughmatris med X-Y-elektromagnet (figur 2.4) som referens

Chalmers		Pughmatris 2 (Relativ beslutsmatris):							
Utfördare:									
Frederic Furborg	Herman Quint	Oskar Hulthén							
Örjan Landgren	Helena Petterson								
Kriterier		Alternativ							
	Vikt	XY-Magnet	r-θ -Magnet	Klomaskin	Robotarm	Små radiostyrda pjäser	Drönare	Elektromagneter i rutnät	Vipp-labyrint
Förflyttningshastighet	5	0		-1	0	0	1	1	-1
Precision	2	0		0	0	-1	-1	-1	-1
Enkelt att programmera	3	1		1	0	-1	-1	-1	-1
Enkel tillverkning	2	1		1	0	-1	1	-1	-1
Vikt	2	0		-1	1	-1	1	-1	-1
Storlek	2	0		-1	1	1	-1	0	-1
Multi-movement	3	-1		-1	0	1	1	1	-1
Användarupplevelse	5	0		-1	-1	1	1	1	1
Ljudnivå	2	0		0	0	0	-1	1	1
Kostnad	1	0		0	0	-1	-1	-1	-1
Nettovärde	27	1	0	-3	1	-2	0	-1	-6
Viktat nettovärde		2	0	-12	-1	0	7	5	-13
Rangordning		3				3	1	2	

Figur A.3: Pughmatris med r-θ-elektromagnet (figur 2.5) som referens

Chalmers		Pughmatris 3 (Relativ beslutsmatris):							
Utfärdare:									
Frederic Furborg	Herman Quint	Oskar Hulthén							
Örjan Landgren	Helena Petterson								
Skapad: 170216									
Modifierad: 170221									
Kriterier	Alternativ								
	Vikt	XY-Magnet	r-θ -Magnet	Klomaskin	Robotarm	Små radiostyrda pjäser	Drönare	Elektromagnet er i rutnät	Vipp-labyrint
Förflyttningshastighet	5	1	1		0	1	1	1	-1
Precision	2	0	0		0	-1	-1	-1	-1
Enkelt att programmera	3	0	-1		-1	-1	-1	-1	-1
Enkel tillverkning	2	1	-1		-1	-1	1	-1	-1
Vikt	2	1	1		0	1	1	-1	-1
Storlek	2	1	1		0	1	-1	1	1
Multi-movement	3	0	1		1	1	1	1	0
Användarupplevelse	5	1	1		1	1	1	1	1
Ljudnivå	2	0	0		0	0	-1	1	0
Kostnad	1	0	0		-1	-1	-1	-1	-1
Nettovärde	27	5	3	0	-1	1	0	0	-4
Viktat nettovärde		16	12	0	2	9	7	7	-8
Rangordning			1	2			3		

Figur A.4: Pughmatris med klomaskin (figur 2.6) som referens

Chalmers		Pughmatris 4 (Relativ beslutsmatris):							
Utfärdare:									
Frederic Furborg	Herman Quint	Oskar Hulthén							
Örjan Landgren	Helena Petterson								
Skapad: 170216									
Modifierad: 170221									
Kriterier	Alternativ								
	Vikt	XY-Magnet	r-θ -Magnet	Klomaskin	Robotarm	Små radiostyrda pjäser	Drönare	Elektromagnet er i rutnät	Vipp-labyrint
Förflyttningshastighet	5	-1	0	0		0	1	1	-1
Precision	2	0	0	0		0	-1	-1	-1
Enkelt att programmera	3	1	0	1		-1	-1	-1	-1
Enkel tillverkning	2	1	0	1		-1	1	-1	0
Vikt	2	1	-1	0		1	1	-1	0
Storlek	2	1	-1	0		1	-1	0	0
Multi-movement	3	-1	0	-1		1	1	1	-1
Användarupplevelse	5	1	1	-1		1	1	1	1
Ljudnivå	2	0	0	0		1	-1	1	0
Kostnad	1	0	0	1		-1	-1	-1	0
Nettovärde	27	3	-1	1	0	2	0	-1	-3
Viktat nettovärde		6	1	-2	0	8	7	5	-8
Rangordning			3			1	2		

Figur A.5: Pughmatris med robotarm (figur 2.3) som referens

Chalmers		Pughmatris 5 (Relativ beslutsmatris):							
Utfärdare:									
Fredric Furborg		Herman Quint	Oskar Hulthén			Skapad: 170216			
Örjan Landgren		Helena Petterson				Modifierad: 170221			
Kriterier	Alternativ								
	Vikt	XY-Magnet	r-θ -Magnet	Klomaskin	Robotarm	Små radiostyrda pjäser	Drönare	Elektromagneter i rutnät	Vipp-labyrint
Förflyttningshastighet	5	0	0	-1	0		1	1	-1
Precision	2	1	1	1	0		-1	-1	-1
Enkelt att programmera	3	1	1	1	1		-1	-1	-1
Enkel tillverkning	2	1	1	1	1		1	1	1
Vikt	2	1	1	-1	-1		1	-1	-1
Storlek	2	-1	-1	-1	-1		-1	-1	-1
Multi-movement	3	-1	-1	-1	-1		0	0	-1
Användarupplevelse	5	-1	-1	-1	-1		1	1	1
Ljudnivå	2	0	0	0	-1		-1	1	1
Kostnad	1	1	1	1	1		-1	1	1
Nettovärde	27	2	2	-1	-2	0	-1	1	-2
Viktat nettovärde		0	0	-9	-8	0	4	6	-7
Rangordning			3	3			2	1	
Vidareutveckling									
Beslut									

Figur A.6: Pughmatris med små radiostyrda pjäser (figur 2.9) som referens

Chalmers		Pughmatris 6 (Relativ beslutsmatris):							
Utfärdare:									
Fredric Furborg		Herman Quint	Oskar Hulthén			Skapad: 170216			
Örjan Landgren		Helena Petterson				Modifierad: 170221			
Kriterier	Alternativ								
	Vikt	XY-Magnet	r-θ -Magnet	Klomaskin	Robotarm	Små radiostyrda pjäser	Drönare	Elektromagneter i rutnät	Vipp-labyrint
Förflyttningshastighet	5	-1	-1	-1	-1	-1		1	-1
Precision	2	1	1	1	1	1		1	-1
Enkelt att programmera	3	1	1	1	1	1		1	1
Enkel tillverkning	2	-1	-1	-1	-1	-1		-1	-1
Vikt	2	-1	-1	-1	-1	-1		-1	-1
Storlek	2	1	1	1	1	1		1	1
Multi-movement	3	-1	-1	-1	-1	0		0	-1
Användarupplevelse	5	-1	-1	-1	-1	-1		-1	-1
Ljudnivå	2	1	1	1	1	1		1	1
Kostnad	1	1	1	1	1	1		1	1
Nettovärde	27	0	0	0	0	1	0	3	-2
Viktat nettovärde		-7	-7	-7	-7	-4	0	6	-11
Rangordning						3	2	1	
Vidareutveckling									
Beslut									

Figur A.7: Pughmatris med drönare (figur 2.10) som referens

Chalmers		Pughmatrix 7 (Relativ beslutsmatrix):							
Utfördare:									
Frederic Furborg	Herman Quint	Oskar Hulthén							
Örjan Landgren	Helena Petterson								
Kriterier		Alternativ							
	Vikt	XY-Magnet	r-θ-Magnet	Klomaskin	Robotarm	Små radiostyrda pjäser	Drönare	Elektromagneter i rutnät	Vipp-labyrint
Förflyttningshastighet	5	-1	-1	-1	-1	-1	-1		-1
Precision	2	1	1	1	1	1	-1		-1
Enkelt att programmera	3	1	1	1	1	1	-1		-1
Enkel tillverkning	2	1	1	1	1	-1	1		-1
Vikt	2	1	1	1	1	1	1		0
Storlek	2	0	0	-1	0	1	-1		-1
Multi-movement	3	-1	-1	-1	-1	0	0		-1
Användarupplevelse	5	-1	-1	-1	-1	-1	1		-1
Ljudnivå	2	-1	-1	-1	-1	-1	-1		0
Kostnad	1	1	1	1	1	-1	-1		-1
Nettovärde	27	1	1	0	1	-1	-3	0	-8
Viktat nettovärde		-5	-5	-7	-5	-6	-6	0	-23
Rangordning			2	2		2	3	3	1
Vidareutveckling									
Beslut									

Figur A.8: Pughmatrix med elektromagneter i rutnät (figur 2.11) som referens

Chalmers		Pughmatrix 8 (Relativ beslutsmatrix):							
Utfördare:									
Frederic Furborg	Herman Quint	Oskar Hulthén							
Örjan Landgren	Helena Petterson								
Kriterier		Alternativ							
	Vikt	XY-Magnet	r-θ-Magnet	Klomaskin	Robotarm	Små radiostyrda pjäser	Drönare	Elektromagneter i rutnät	Vipp-labyrint
Förflyttningshastighet	5	1	1	1	1	1	1	1	
Precision	2	1	1	1	1	1	1	1	
Enkelt att programmera	3	1	1	1	1	1	-1	1	
Enkel tillverkning	2	1	1	1	0	-1	1	1	
Vikt	2	1	1	1	0	1	1	0	
Storlek	2	1	1	-1	0	1	-1	1	
Multi-movement	3	0	1	0	1	1	1	1	
Användarupplevelse	5	-1	-1	-1	-1	-1	1	1	
Ljudnivå	2	1	-1	0	0	-1	-1	0	
Kostnad	1	1	1	1	0	-1	-1	1	
Nettovärde	27	7	6	4	3	2	2	8	0
Viktat nettovärde		14	13	8	8	7	11	23	0
Rangordning			2	3				1	
Vidareutveckling									
Beslut									

Figur A.9: Pughmatrix med vipp-labyrint (figur 2.13) som referens